

WORLD WIDE WEB NEWS RETRIEVAL SYSTEM

Patent number: JP10254912
Publication date: 1998-09-25
Inventor: NEHAB SMADAR; WICKRAMARATNE MANJULA G;
KLARK PAUL L
Applicant: CANON INFORMATION SYST INC
Classification:
- international: G06F17/30; G06F12/00; G06F13/00
- european: G06F17/30W1F
Application number: JP19970307770 19971006
Priority number(s): US19960726853 19961004

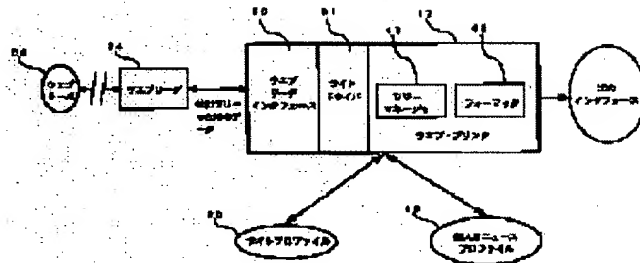
Also published as:

EP0834822 (A2)
US6029182 (A1)
EP0834822 (A3)

Report a data error here

Abstract of JP10254912

PROBLEM TO BE SOLVED: To naturally read the data by flattening the data tree where the data extracted from a hypermedia document are built into a linear document and then turning this document into a format. **SOLUTION:** A web printer 17 accesses various web sites and gives a command to a web reader 34 to secure the connection to a web via a web server 35 and based on the information stored in a personal news profile 19 and a site profile 20, so that the data are retrieved from those web sites. The reader 34 sends the retrieved data to the printer 17, and the printer 17 uses the received data to assemble an extracted data tree. Then the printer 17 flattens the assembled data tree to converts it into a linear document and then turns this document into an output format via an output interface.



Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-254912

(43) 公開日 平成10年(1998) 9月25日

(51) Int.Cl.⁹

識別記号

F I

G 0 6 F 17/30

G 0 6 F 15/403

3 8 0 Z

12/00

5 4 7

12/00

5 4 7 H

13/00

3 5 1

13/00

3 5 1 G

15/40

3 1 0 F

15/419

3 2 0

審査請求 未請求 請求項の数54 O L 外国語出願 (全116頁)

(21) 出願番号 特願平9-307770

(22) 出願日 平成9年(1997)10月6日

(31) 優先権主張番号 08/726853

(32) 優先日 1996年10月4日

(33) 優先権主張国 米国 (US)

(71) 出願人 592208172

キヤノン インフォメーション システム
ズ インク.Canon Information S
ystems, Inc.

アメリカ合衆国 カリフォルニア州

92612, アーバイン, イノベーション ド
ライブ 110

(72) 発明者 スメーダー ネハブ

アメリカ合衆国 カリフォルニア州

94306, パロ アルト, ウィルキー

ウエイ 4020

(74) 代理人 弁理士 大塚 康德 (外1名)

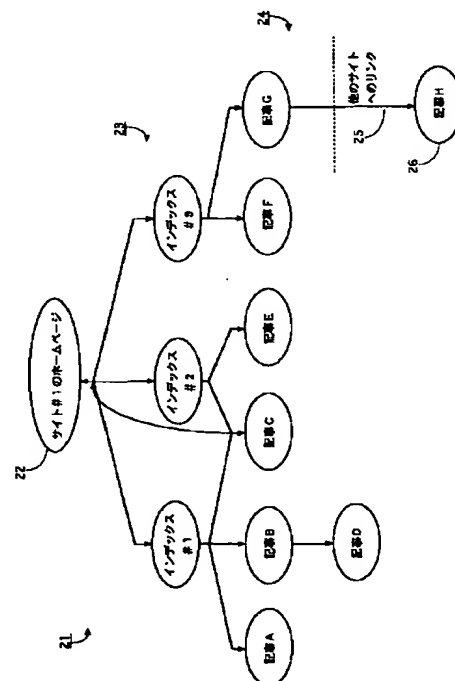
最終頁に続く

(54) 【発明の名称】 ワールドワイド・ウェブ・ニュース検索システム

(57) 【要約】 (修正有)

【課題】 目的とするニュースを適確容易にアクセスする。

【解決手段】 (1) ウェブ・リーダーを起動しワールド・ワイド・ウェブに接続し、(2) ウェブサイトアドレス情報及びウェブサイトコマンドを検索して取り出し、(3) ウェブサイトにアクセスするように、ウェブ・リーダーに命令し、(4) ループを形成するハイパー・メディア・リンクを避け、リンクされたリストを参照して前記ウェブサイトコマンドに基づいてウェブサイトからウェブサイトデータをダウンロードし、(5) ウェブサイトデータを線形ドキュメントに格納し、(6) 格納されたウェブサイトアドレス情報にあるアドレスすべてがアクセスされてしまうまで、(1) ~ (5) の処理を繰り返し、フォーマット情報に基づいて線形ドキュメントを個人化ドキュメントの形にフォーマットする。



【特許請求の範囲】

【請求項1】 少なくとも1つのハイパー・メディア・ドキュメントからデータをフォーマットする方法であって、

少なくとも1つのハイパー・メディア・ドキュメントにアクセスするアクセス工程と、

前記ハイパー・メディア・ドキュメントからデータを、前記ハイパー・メディア・ドキュメントの構造に基づいて、抽出データ・ツリーへと取り出す取出し工程と、この抽出データ・ツリーを線形ドキュメントに平坦化するための平坦化工程と、

その線形ドキュメントをフォーマット化されたドキュメントにフォーマットするためのフォーマット工程とを具備することを特徴とするデータフォーマット方法。

【請求項2】 そのフォーマットされたドキュメントをプリントする工程をさらに具備することを特徴とする請求項1に記載のデータフォーマット方法。

【請求項3】 前記ハイパー・メディア・ドキュメントはワールド・ワイド・ウェブに置かれることを特徴とする請求項1に記載のデータフォーマット方法。

【請求項4】 前記ハイパー・メディア・ドキュメントはインターネットに置かれることを特徴とする請求項1に記載のデータフォーマット方法。

【請求項5】 前記ハイパー・メディア・ドキュメントはイントラネットに置かれることを特徴とする請求項1に記載のデータフォーマット方法。

【請求項6】 前記アクセス工程、前記取出し工程、前記平坦化工程および前記フォーマット工程は個人用ニュース・プロフィールに従って実行されることを特徴とする請求項1に記載のデータフォーマット方法。

【請求項7】 ハイパー・メディア・リンクのコンピュータネットワークからデータを抽出するための個人用ニュース・プロフィールを創成する創成方法であって、ハイパー・メディア・リンクのコンピュータネットワークにアクセスする工程と、学習モードを入力する工程と、

コマンドによって前記ハイパー・メディア・リンクのコンピュータネットワーク上のサイトを横断する工程と、

そのコマンドから少なくとも1つの規則を抽出する工程と、

その少なくとも1つの規則を個人用ニュース・プロフィールにまとめるコンパイル工程とを具備することを特徴とするプロフィール創成方法。

【請求項8】 前記少なくとも1つの規則は、前記ハイパー・メディア・リンクのコンピュータネットワークを横断するために、サイトの構造的な特徴を指定することを特徴とする請求項7のプロファイル創成方法。

【請求項9】 前記少なくとも1つの規則は、前記ハイパー・メディア・リンクのコンピュータネットワーク

を横断するために、内容に基づく基準を指定することを特徴とする請求項8のプロファイル創成方法。

【請求項10】 ウェブサイト検索データ検索システムのための個人化プロフィールを創成するための個人化システムであって、

ワールド・ワイド・ウェブにアクセスするために、データおよびコマンドを入力するための入力部と、前記ワールド・ワイド・ウェブとの接続部と、

前記個人化システムからのコマンドに従って、前記ワールド・ワイド・ウェブとの前記接続部を介して前記ワールド・ワイド・ウェブにアクセスするためのウェブ・リーダを格納するためのメモリと、

ユーザのコマンドに応答して個人化システムを起動するためのプロセッサとを具備し、

前記個人化システムは、起動されると、(1)ウェブ・リーダを起動し、(2)ウェブ・リーダを通じてワールド・ワイド・ウェブにアクセスし、(3)学習モードを入力し、(4)ユーザコマンドに従ってワールド・ワイド・ウェブを横断するように前記ウェブ・リーダに命令を送り、(5)そのユーザコマンドから少なくとも1つの規則を抽出し、(6)その少なくとも1つの規則を個人化プロフィールにまとめ、(7)その個人化プロフィールを格納することを特徴とする個人化システム。

【請求項11】 ハイパー・メディア・リンクのコンピュータネットワークから記事を検索し、その記事を個人化された新聞の形にフォーマットするための方法であって、前記ハイパー・メディア・リンクのコンピュータネットワーク上のサイトを示すアドレスデータと、そのサイトからデータにアクセスするためのコマンドデータと、新聞レイアウトコマンドとを有する格納された個人用ニュース・プロフィールを取り出す工程と、その個人用ニュース・プロフィールに格納されるアドレスデータに基づいて前記サイトに連絡をとる工程と、前記個人用ニュース・プロフィールに格納されるコマンドデータに基づいてそのサイトから記事をダウンロードするための工程と、その記事を線形ドキュメントの形に平坦化する工程と、前記線形ドキュメントを、前記個人用ニュース・プロフィールに格納されたレイアウトコマンドに従って個人化された新聞の形にフォーマットする工程とからを具備することを特徴とする記事記事取出しフォーマット方法。

【請求項12】 その個人化された新聞をプリントする工程を具備することを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項13】 前記ハイパー・メディア・リンクのコンピュータネットワークはワールド・ワイド・ウェブであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項14】 前記ハイパー・メディア・リンクのコンピュータネットワークはイントラネットであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項15】 前記ハイパー・メディア・リンクのコンピュータネットワークはインターネットであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項16】 前記ハイパー・メディア・リンクのコンピュータネットワークはイントラネットであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項17】 前記ハイパー・メディア・リンクのコンピュータネットワークはインターネットであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

【請求項18】 前記ハイパー・メディア・リンクのコンピュータネットワークはイントラネットであることを特徴とする請求項11に記載の記事取出しフォーマット方法。

ンピュータネットワークはインターネット上にあることを特徴とする請求項 1 に記載の記事取出しフォーマット方法。

【請求項 15】 前記ハイパー・メディア・リンクのコンピュータネットワークはイントラネット上にあることを特徴とする請求項 1 に記載の記事取出しフォーマット方法。

【請求項 16】 データにアクセスするための前記コマンドデータは、サイトの構造に基づいて記事を選択するためのデータを含むことを特徴とする請求項 1 に記載の記事取出しフォーマット方法。

【請求項 17】 データにアクセスするための前記コマンドデータは、記事の内容に基づいて記事を選択するためのデータを含むことを特徴とする請求項 16 に記載の記事取出しフォーマット方法。

【請求項 18】 少なくとも 1 つのウェブサイトにアクセスし、そのウェブサイトからデータを検索して取り出し、そのデータを個人化されたドキュメントの形にフォーマットする、ワールド・ワイド・ウェブ・サイト・データ検索システムであって、

ワールド・ワイド・ウェブにアクセスするためにデータおよびコマンドを入力するための入力部と、ウェブ・リーダを含むウェブ・サイト・データ検索ドライバと、格納されたウェブサイトアドレス情報と、格納されたフォーマット情報とを格納するためのメモリであって、ウェブサイトに接続し、接続されたウェブサイトの中でコマンドを発行する工程を含むことを特徴とするメモリと、

ワールド・ワイド・ウェブと接続する接続部と、ワールド・ワイド・ウェブにアクセスするためのコマンドを入力するユーザにตอบสนองしてウェブ・サイト・データ取り出しドライバを起動するプロセッサであって、ウェブ・サイト・データ取り出しドライバは起動されると、

(1) ウェブ・リーダを起動して前記接続部を介してワールド・ワイド・ウェブに接続し、(2) ウェブサイトアドレス情報およびウェブサイトコマンドを検索して取り出し、(3) このウェブサイトアドレス情報およびウェブサイトコマンドに基づいて、ウェブサイトにアクセスするように、ウェブ・リーダに命令し、(4) ループを形成するハイパー・メディア・リンクを避け、そして既にダウンロードされたデータを繰り返しダウンロードすることを避けるために、リンクされたリストを参照して前記ウェブサイトコマンドに基づいてウェブサイトからウェブサイトデータをダウンロードし、(5) そのウェブサイトデータを線形ドキュメントに格納し、(6) 格納されたウェブサイトアドレス情報にあるアドレスすべてがアクセスされてしまうまで、工程 1 から工程 5 まですを繰り返し、前記フォーマット情報に基づいて前記線形ドキュメントを個人化ドキュメントの形にフォーマットすることを特徴とするワールド・ワイド・ウェブ・サ

イト・データ検索システム。

【請求項 19】 前記ウェブサイトアドレス情報、ウェブサイトコマンドおよびメモリに格納されたフォーマット情報は個人用ニュース・プロフィールを形成することを特徴とする請求項 18 に記載のウェブサイトデータ検索システム。

【請求項 20】 更に個人化されたドキュメントをプリントするためのプリンタを具備することを特徴とする請求項 18 に記載のウェブサイトデータ検索システム。

【請求項 21】 個人化されたドキュメントは個人化された新聞の型を示すことを特徴とする請求項 18 に記載のウェブサイトデータ検索システム。

【請求項 22】 個人化されたドキュメントは個人化された雑誌の型を示すことを特徴とする請求項 18 に記載のウェブサイトデータ検索システム。

【請求項 23】 個人化されたドキュメントは個人化された書籍の型を示すことを特徴とする請求項 18 に記載のウェブサイトデータ検索システム。

【請求項 24】 コンピュータ読み取り媒体に格納されたコンピュータ実行処理工程であって、ワールド・ワイド・ウェブサイトにアクセスし、サイトのデータを検索し、そのデータを個人化されたドキュメントの形にフォーマットする実効処理工程であって、

ワールド・ワイド・ウェブに接続する接続工程と、ユーザ定義のウェブサイトアドレス情報、ユーザ定義のウェブサイトコマンド、およびユーザ定義のフォーマットコマンドを取り出すための取り出し工程と、ユーザ定義の前記ウェブサイトアドレス情報に基づいてウェブサイトアクセスし、ユーザ定義の前記ウェブサイトコマンドに基づいてウェブサイト内からデータを検索するように、ウェブ・リーダを起動させるための起動工程と、アクセスされた前記ウェブサイトから取り出された前記ウェブサイトデータを、抽出データ・ツリーの形でダウンロードするためのダウンロード工程と、前記抽出データ・ツリーを線形ドキュメントの形に平坦化するための平坦化工程と、

ユーザ定義の前記ウェブサイトアドレス情報にある全てのアドレスがアクセスされてしまうまで、ダウンロード工程および平坦化工程を繰り返すための工程と、ユーザ定義の前記フォーマットコマンドに従って、前記格納されたデータを個人化されたドキュメントの形にフォーマットするためのフォーマット工程とを具備することを特徴とするコンピュータ実行可能処理工程。

【請求項 25】 前記個人化されたドキュメントを出力装置へスプールするためのスプール工程を具備することを特徴とする請求項 24 に記載のコンピュータ実行可能処理工程。

【請求項 26】 その出力装置はプリンタであることを特徴とする請求項 25 に記載のコンピュータ実行可能処理工程。

【請求項27】 その出力装置はディスプレイであることを特徴とする請求項25に記載のコンピューター実行可能処理工程。

【請求項28】 ユーザ定義のウェブサイトコマンドはウェブサイトの情報に基づいてデータを選択するためのコマンドを含むことを特徴とする請求項24に記載のコンピューター実行可能処理工程。

【請求項29】 ユーザ定義のウェブサイトコマンドはウェブサイトの内容に基づいてデータを選択するためのコマンドを含むことを特徴とする請求項28に記載のコンピューター実行可能処理工程。

【請求項30】 ワールド・ワイド・ウェブ上のオンラインニュースサービスからニュース記事を検索して取出し、そのニュース記事を個人化された新聞の形にフォーマットするためのニュース検索装置であって、

(1) ウェブサイトからのデータにアクセスするためのアドレスデータおよびコマンドデータと(2) 新聞フォーマットコマンドとを格納する第1の格納手段と、格納された前記個人用ニュース・プロフィールを検索してそこに格納されたデータにアクセスするための検索取出し手段と、

前記個人用ニュース・プロフィールに格納されたアドレスデータに基づいて、ウェブサイトへ接続するべくウェブ・リーダを起動する起動手段と、

前記個人用ニュース・プロフィールに格納されたコマンドデータに基づいて、接続されたウェブサイトからニュース記事をダウンロードするダウンロード手段、ダウンロードされた前記ニュース記事を格納する第2の格納手段と、

前記個人用ニュース・プロフィールに格納された新聞フォーマット・コマンドに基づいて、格納された前記ニュース記事を個人化された新聞の形にフォーマットするためのフォーマット手段とを具備することを特徴とするニュース検索装置。

【請求項31】 更に、個人化された新聞をプリンタヘスプールするためのスプール手段を具備した請求項30に記載のニュース検索装置。

【請求項32】 データをハイパー・メディア・ドキュメントから個人化されたドキュメントにフォーマットするデータフォーマット方法であって、

前記ハイパー・メディア・ドキュメントの位置を指定するための位置指定工程と、

前記ハイパー・メディア・ドキュメントの型式を指定するための型式指定工程と、

ハイパー・メディア・ドキュメントの構造に基づいて決められた範囲であって、前記ハイパー・メディア・ドキュメントから検索するべきデータの範囲を指定する範囲指定工程と、

前記ハイパー・メディア・ドキュメントから取出されたデータを個人化されたドキュメントの形にフォーマット

するためのフォーマットを指定するフォーマット指定工程と、

前記位置指定工程で指定された位置で見つけられたハイパー・メディア・ドキュメントにアクセスするアクセス工程と、

前記型式指定工程で指定された形式と前記範囲指定工程で指定された範囲とに従ってデータを検索すべく、前記アクセス工程でアクセスされたハイパー・メディア・ドキュメントから前記データを検索する検索工程と、

10 前記フォーマット指定工程で指定されたフォーマットに従ってそのデータをフォーマットすべく、前記検索工程で検索された前記データを個人化されたドキュメントの形にフォーマットするフォーマット工程とを具備することを特徴とするデータフォーマット方法。

【請求項33】 更に、個人化されたドキュメントをプリントするためのプリント工程を具備することを特徴とする請求項32に記載のデータフォーマット方法。

【請求項34】 前記位置指定工程で指定される位置はファイル名であることを特徴とする請求項32に記載のデータフォーマット方法。

【請求項35】 前記位置指定工程で指定される位置は、ワールド・ワイド・ウェブのためのユニフォーム・リソース・ロケータであることを特徴とする請求項32に記載のデータフォーマット方法。

【請求項36】 ハイパー・メディア・ドキュメントを処理する処理方法であって、

前記ハイパー・メディア・ドキュメントにアクセスする工程と、

前記前記ハイパー・メディア・ドキュメントからアドレスを抽出する工程と、

前記ハイパー・メディア・ドキュメントから抽出されたアドレスをメモリの中のコンテナに格納する工程と、

前記コンテナに格納されたアドレスに格納されたデータを処理するための処理機能を起動する工程と、

前記コンテナ内のアドレスに格納されたデータを前記メモリへダウンロードする工程と、

予め決定された構成情報に従って、ダウンロードされた前記データから所定のデータを抽出する工程と、

40 予め定義されたフォーマット設定に従って、前記所定のデータをフォーマットし、フォーマットされたドキュメントを生成するフォーマットする工程と、

前記処理機能に従ってフォーマットされたドキュメントを処理する工程とを具備することを特徴とするドキュメント処理方法。

【請求項37】 更に、フォーマットされたドキュメントを処理する前に、フォーマットされたドキュメントをプレビューする工程を具備することを特徴とする請求項36に記載のドキュメント処理方法。

【請求項38】 更に、ドキュメントをプレビューした後であって、フォーマットされたドキュメントを処理機

能に従って処理する前において、フォーマットの設定を変更する工程と、

前記処理機能を再起動する工程と、

フォーマットされたドキュメントを生成するために、変更されたフォーマットの設定に従ってデータを再フォーマットする工程とを具備することを特徴とする請求項37に記載のドキュメント処理方法。

【請求項39】 前記アドレスが前記コンテナに入力される順序に従って、前記アドレスが前記コンテナに格納され、

そのアドレスがコンテナに格納される順序に従って、予め決定されるデータを処理機能が処理することを特徴とする請求項36に記載のドキュメント処理方法。

【請求項40】 更に、コンテナ内のアドレスをドラッグおよびドロップすることにより、コンテナに格納されるアドレスを再配列する工程を具備することを特徴とする請求項39に記載のドキュメント処理方法。

【請求項41】 更に、フォーマットの設定および構成情報をグラフィック・ユーザ・インターフェースを通して入力する工程を具備することを特徴とする請求項36に記載のドキュメント処理方法。

【請求項42】 グラフィック・ユーザ・インターフェースは、その1つが処理機能を起動する複数の処理アイコンを有することを特徴とする請求項41に記載のドキュメント処理方法。

【請求項43】 グラフィック・ユーザ・インターフェースは複数のモードで表示されることを特徴とする請求項42に記載のドキュメント処理方法。

【請求項44】 前記複数のモードは、(1)前記グラフィック・ユーザ・インターフェースが、フォーマットするフィールドと、処理オプションと、メニューと、上記処理のアイコンとを表示するフル機能モードと、

(2)前記グラフィック・ユーザ・インターフェースが前記処理アイコンのみを表示する最小化モードとを有することを特徴とする請求項43に記載のドキュメント処理方法。

【請求項45】 前記最小化モードにあるグラフィック・ユーザ・インターフェースは、ハイパー・メディア・ドキュメントをブラウジングする間に表示されることを特徴とする請求項44に記載のドキュメント処理方法。

【請求項46】 ハイパー・メディア・ドキュメントを処理する処理装置であって、前記ハイパー・メディア・ドキュメントにアクセスするウェブ・リーダと、

前記ハイパー・メディア・ドキュメントからアドレスを抽出する手段と、

前記ハイパー・メディア・ドキュメントから抽出されたアドレスを格納するコンテナを含むメモリと、

コンテナに格納されたアドレスに格納されたデータを処理するために、少なくとも1つの処理機能を起動する処

理アイコンを有するグラフィック・ユーザ・インターフェースと、

(1)前記コンテナに格納されたアドレスに格納されたデータをメモリへダウンロードし、(2)予め定義された構成の設定に従って、ダウンロードされた前記データから所定のデータを抽出し、(3)フォーマットされたドキュメントを生成させるための予め定義されたフォーマットの設定に従って前記所定のデータをフォーマットし、(4)前記処理機能に従ってフォーマットされたドキュメントを処理する処理手段と、を具備することを特徴とするドキュメント処理装置。

【請求項47】 更に、フォーマットされたドキュメントを処理する前に、フォーマットされたドキュメントをプレビューするプレビュー手段とを具備することを特徴とする請求項46に記載のドキュメント処理装置。

【請求項48】 アドレスがコンテナに入力された順序に従って前記アドレスがコンテナに格納され、前記アドレスが前記コンテナに格納された順序に従って、前記所定のデータを前記処理機能が処理することを特徴とする請求項46に記載のドキュメント処理装置。

【請求項49】 更に、前記コンテナにあるアドレスを再配列するために、前記コンテナでリストされたアドレスをドラッグおよびドロップするためのドラッグ・ドロップ手段を具備することを特徴とする請求項48に記載のドキュメント処理装置。

【請求項50】 更に、フォーマットの設定および構成情報をグラフィック・ユーザ・インターフェースを通して入力するための入力手段を具備することを特徴とする請求項46に記載のドキュメント処理装置。

【請求項51】 グラフィック・ユーザ・インターフェースは、その1つが処理機能を起動する複数の処理アイコンを具備することを特徴とする請求項50に記載のドキュメント処理装置。

【請求項52】 グラフィック・ユーザ・インターフェースを複数のモードで表示するための表示手段を具備することを特徴とする請求項51に記載のドキュメント処理装置。

【請求項53】 前記複数のモードは、

(1)前記グラフィック・ユーザ・インターフェースが、フォーマットするフィールドと、処理オプションと、メニューと、上記処理のアイコンとを表示するフル機能モードと、(2)前記グラフィック・ユーザ・インターフェースが前記処理アイコンのみを表示する最小化モードとを具備することを特徴とする請求項52に記載のドキュメント処理装置。

【請求項54】 最小化モードで表示されたグラフィック・ユーザ・インターフェースは、前記ハイパー・メディア・ドキュメントをブラウジングする間に表示されることを特徴とする請求項53に記載のドキュメント処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・ネットワーク上のハイパー・メディア・ドキュメントを自動的に横断して、これらドキュメントから、そのドキュメントの構造とパーソナライズ(personalized)されたデータ検索構造との合致に基づいて、情報を自動的に検索するデータ検索システムに関する。より詳しくいうと、本発明は、ワールド・ワイド・ウェブ上の、あるいはハイパー・メディア・リンクをサポートする私的なコンピュータネットワーク上の、或いは、他のあらゆるハイパー・メディアでリンクされたコンピュータシステム上の、ニュースサービスから、あるいはマガジンサービスから、または両者を組み合わせたものから、記事を検索することができる。

【0002】たとえば、ニューヨークタイムズ(New York Times)からニュース記事を検索するためのウェブサイトや、雑誌ピープル(People)から記事を検索するウェブ・サイトが存在する。本発明の検索システムは、このようなウェブサイトを横断し、パーソナライズされたデータ検索構造に基づいて記事を選択することができる。この個人化データ検索構造は、第1面のみの、ビジネス欄のヘッディングの、株式欄の、スポーツ欄等のヘッディングの全文を検索するように命令するコマンドを含むこともできる。さらに、この個人化データ検索構造は、コンテンツ依存型ルール(content-based rules)を含むことができ、あるキーワードを有する記事を検索し、ある類は、あるキーワードを有する記事を除外し、あるいは、ある規則に基づく内容分析に基づいた記事を含むこともできる。本発明は、また、検索したニュース記事すべてを統合し、その統合したニュース記事を、その記事の各々がユーザが予め定義したレイアウトに基づいて配置されるような、新聞型のフォーマットで印刷するための方法を提供する。

【0003】上記の例はWWウェブに関連するものだが、ハイパー・メディア・ドキュメントはウェブだけでなく、イントラネットのような他の型のネットワークにも存在することができる。イントラネットは外部のコンピュータネットワークに接続しない私的なコンピュータネットワークである。たとえば会社所有のコンピュータネットワークはハイパー・メディア・ドキュメントを内蔵するイントラネットになることができる。要するに、以下の議論はワールド・ワイド・ウェブに関して行われるものである。しかしながら本発明は、ハイパー・メディア・ドキュメントを容れるあらゆるタイプのコンピュータネットワーク（例えば、イントラネット、あるいはインターネット上のWWウェブ以外のハイパー・メディア・リンクのコンピュータ・ネットワーク）にも等しく良好に適合することが理解されよう。

【0004】WWウェブ上のハイパー・メディア・ドキュ

メントは、複数のウェブ・サイトにまたがることができる。このようなドキュメントには、新聞、新聞記事、雑誌、カタログ、マニュアル、覚書、その他同種のものが可能である。要するに、これから行われる議論はニュース情報のソースに関して行われるものである。しかし、本発明はハイパー・メディア・ドキュメントのいずれの型にも、等しく良好に適合することが理解されよう。

【0005】

【従来の技術】ワールド・ワイド・ウェブ(WWウェブ)とはハイパー・メディア・ドキュメントのオンラインのソースであって、ハイパー・メディア・ドキュメントは、ウェブサイトなど、他のドキュメントやウェブサイト等へのリンクとして機能するハイパー・メディア・テキストとハイパー・メディア・イメージとを有する。従って、WWウェブ上のドキュメントは連続的に構成されない。むしろ、ユーザは他のドキュメントまたはウェブ・サイトに自動的にリンクされるのであって、ドキュメント内にあるテキストリンクやイメージリンクのようなハイパー・メディア・リンクを選択することによって、ドキュメント全体を見ることができる。従って、ドキュメント全体はテキストをスクロールしても見ることができない。

【0006】このウェブの一般的な使い方のひとつは、雑誌および新聞のオンラインでの発行および発信である。最近では、ニューヨークタイムズ等の多くのウェブ・ニュース・サービスが、ユーザが関心事についてのキーワードを定め、キーワードに合うテキストを有するニュース情報を時々刻々受け取れるようにしている。それで、ニュース情報はユーザのコンピュータにモデムやEメールを通じて送られる。しかし、ニューヨークタイムズのような、ほとんどのウェブ・ニュース・サイトの新聞は、含む情報が多すぎるために、しかもその情報がキーワードの合致によってのみ検索されるために、そのほとんどがユーザにとって無関係のものである。

【0007】ニュース情報のソースは、そのほか、“インディビジュアル・インクIndividual Inc.”のような情報提供者により提供される。インディビジュアル・インクは、ユーザにより予め定められたキーワードに基づいて、もっとも関連性の強い上位二十の記事の短い要約をユーザに提供する。この予約購読制のニュース・サービスは、ユーザがキーワードに基づいて重要度の範囲を五から十までに仕分けして、それに優先順位をつけることができるようにする。この情報サービスは、それらのキーワードのいずれかを有する雑誌と新聞を求めてWWウェブを探索する。そのキーワード検索に基づいて、もっとも関連性の強い記事のうち二十が選出され、1頁の短い要約にまとめられて、ユーザに提供するためにファクシミリでユーザへ送られる。しかしユーザが、要約ではなくてドキュメント全体を見るためには、そのドキュメントを含む特定のウェブサイトにログオンして、その

ドキュメントを検索しなければならない。

【0008】ユーザが新聞を個人化(personalize)して端末画面に表示させることができるようにするサービスがほかにもまだあり、それは、ウェブ上の種々のニュース・ソースから種々のニュース記事へのリンクを格納することによって可能である。たとえばCRAYON (Create Your Own Newspaper)は、ユーザが、25以上の異なるオンライン新聞へのリンクの中から特定のセクションを選択して、その選択を個人化された新聞に構成することができるようにするものである。CRAYONを使用すれば、たとえばニューヨーク・タイムズの国際関係欄、ウォール・ストリート・ジャーナルのビジネス欄、シカゴ・トリビューンのスポーツ欄へのリンクを有する個人化された新聞を構成することができる。その後、この新聞のためのHTML(hypertext markup language)ソース・ファイルは、後で使用するためのマス・メディア・ストレージに格納される。

【0009】

【発明が解決しようとする課題】前述のニュース・サービスや情報サービスが、ニュースを最新のものに保つための便利な方法を提供するとしても、人が現実世界の新聞を自然に読むような方法で、ユーザがニュースにアクセスできたり、見たりすることはできるようにするものでない。即ち、人は、おもしろいと思う欄のページを走り読みした後で、自分の注意がひかれた記事を読むことによって、自然にニュースを読むのである。換言すると、人は、構造的な入り方をするもので、初めにどのページ(たとえばビジネス欄、世界欄の一ページ目、そして芸術欄のコミックのページ)を読むかを決める。それから、選択したページ内で記事を選択するのである。

【0010】要するに、従来のニュース・情報サービスでは、ユーザは、ハイパー・メディア・ドキュメントからデータに、そのドキュメントの構造に基づいてアクセスすることはできず、さらにユーザが、そのデータを自然にスキャンし読むことができるように、そのデータをフォーマットすることができない。

【0011】

【課題を解決するための手段】本発明は当該技術における上述の問題点の解決に向けられたもので、少なくともひとつのハイパー・メディア・ドキュメントにアクセスし、次に、そのハイパー・メディア・ドキュメントからデータを抜き出してデータ・ツリーに組み入れ(この場合、そのデータは、前記ハイパー・メディア・ドキュメントの構造に基づいて検索される)、さらにその抜き出されたデータ・ツリーを線形(linear)のドキュメントの形に平坦化し、そしてその線形のドキュメントを定型のドキュメントの形にフォーマットすることによりなされる。

【0012】別の面において、本発明は、ハイパー・メディア・リンクのコンピュータ・ネットワークからデー

タを検索するための、個人用ニュース・プロフィールを作るものである。まず、そのハイパー・メディア・リンクのコンピュータネットワークがアクセスされ、学習モードが開始され、ハイパー・メディア・リンクのコンピュータネットワークはコマンドで横断され、少なくとも1個の規則がそのコマンドから抜き出され、そしてその規則は個人用ニュース・プロフィールの形にまとめられる。

【0013】さらに別の面において、本発明はウェブサイト検索・データ検索システムのための、個人化プロフィールを作るものである。データとモードがワールド・ワイド・ウェブにアクセスするために入力され、ワールド・ワイド・ウェブとのコネクションが作られる。ウェブ・リーダが起動され、このウェブ・リーダがコネクションを通して、ウェブにアクセスする。ユーザコマンドに応答して、学習モードが開始される。ワールド・ワイド・ウェブをトラバースするためにコマンドが送られ、このコマンドから、少なくともひとつの規則が抽出される。この規則は個人化プロフィールにまとめられ、この個人化ファイルは格納される。

【0014】さらに別の面において、本発明は、ハイパー・メディアにリンクしたコンピュータネットワークから記事を検索して、その記事を個人化された新聞にフォーマットするものである。格納された個人用ニュース・プロフィールが検索される。この個人用ニュース・プロフィールはハイパー・メディア・リンクのコンピュータネットワーク上のサイトのアドレスデータ、このサイトからデータをアクセスするためのコマンドデータ、および新聞レイアウトコマンドを含む。このサイトは個人用ニュース・プロフィールに格納されるアドレスデータに基づいてアクセスされ、そしてサイトの記事は、個人用ニュース・プロフィールに格納されるコマンド・データに基づいてダウンロードされる。そのダウンロードされた記事は線形ドキュメントの形に平坦化され、この線形ドキュメントは、個人用ニュース・プロフィールに格納される新聞・レイアウト・コマンドに従って、個人化された新聞の形にフォーマットされる。

【0015】さらに別の面において、本発明は、ワールド・ワイド・ウェブのサイトからデータを検索し、そのデータを、個人化されたドキュメントの形にフォーマットするものである。ウェブ・リーダと、格納されたウェブサイト・アドレス情報と、格納されたウェブサイト・コマンドと、格納されたフォーマット情報とを有するウェブ・サイト・データ検索ドライバーがアクセスされる。本発明は、(1)ワールド・ワイド・ウェブに、このウェブとのコネクションを通じて接続するために、ウェブ・リーダを開始し、(2)ウェブサイト・アドレス情報ならびにウェブサイト・コマンドを検索し、(3)ウェブサイト・アドレス情報ならびにウェブサイト・コマンドに基づいてウェブ・サイトにアクセスするため

に、ウェブ・リーダに命令し、(4) ループを形成するハイパー・メディア・リンクを避けるように、そしてダウンロード済みのデータを繰返しダウンロードしないように、リンクされたリストに参照してデータをダウンロードするように命令するウェブサイト・コマンドに基づいて、ウェブサイトからウェブサイト・データをダウンロードし、(5) そのウェブサイト・データを線形ドキュメントに格納し、(6) 格納されたウェブサイト・アドレス情報にあるアドレスがすべてアクセスされてしまうまで、工程(2) から工程(5) までを繰返し、そして(7) フォーマット情報に基づいて、線形ドキュメントを個人化されたドキュメントの形にフォーマットする。

【0016】さらに別の面において、本発明は、ワールド・ワイド・ウェブ・サイトにあるデータにアクセスし、検索し、そのデータを個人化されたドキュメントの形にフォーマットするものである。本発明はワールド・ワイド・ウェブに接続し、ユーザが定義したウェブサイト・アドレス情報、ユーザが定義したウェブサイト・コマンド、ユーザが定義したフォーマット・コマンドを検索し、このユーザが定義するウェブサイト・アドレス情報に基づいてウェブサイトアクセスするように、ウェブ・リーダを起動する。このウェブ・リーダは、ユーザが定義するウェブ・サイト・コマンドに基づいて、データをウェブからダウンロードするために使われ、そしてそのデータは抽出されたデータ・ツリーのなかへダウンロードされる。ユーザが定義するウェブサイト・アドレス情報にある全てのアドレスがアクセスされてしまうまで、このダウンロードは続く。抽出されたデータ・ツリーは線形ドキュメントの形で平坦化され、そしてこの平坦化されたドキュメントは、ユーザが定義したフォーマット・コマンドに基づいて個人化されたドキュメントの形にフォーマットされる。

【0017】さらに別の面において、本発明は、ワールド・ワイド・ウェブ上のオンライン・ニュース・サービスから、ニュース記事を検索し、そのニュース記事を個人化された新聞の形にフォーマットするものである。本発明は、ウェブサイトからのデータにアクセスするための、アドレス・データおよびコマンド・データ、ならびに新聞フォーマット・コマンドからなる個人用ニュース・プロフィールを格納し、その格納された個人用ニュース・プロフィールを検索して、そこに格納されたデータにアクセスし、個人用ニュース・プロフィールに格納されたアドレス・データに基づいて、ウェブ・サイトにコンタクトするように、ウェブ・リーダを起動し、個人用ニュース・プロフィールに格納されたコマンド・データに基づいて、コンタクトされたウェブ・サイトでニュースをダウンロードし、そのダウンロードされたニュース記事を格納し、その格納された新聞記事を、個人用ニュース・プロフィールに格納された新聞フォーマット・

コマンドに基づいて、個人化された新聞の形にフォーマットする。

【0018】さらに別の面において、本発明は、ハイパー・メディア・ドキュメントを、個人化されたドキュメントの形にフォーマットするものである。ハイパー・メディア・ドキュメントの位置が指定され、ハイパーメディア・ドキュメントの型が指定され、ハイパー・メディア・ドキュメントから検索すべきデータの領域が、ハイパー・メディア・ドキュメントの構造に基づくように指定され、そしてフォーマットが、ハイパー・メディア・ドキュメントから検索されるデータを、個人化されたドキュメントの形にフォーマットするために指定される。その指定された位置で見つけられたハイパー・メディア・ドキュメントはアクセスされ、データは、指定されたハイパー・メディア・ドキュメントの型および指定された範囲に従って、ハイパー・メディア・ドキュメントから検索され、そしてそのデータは指定されたフォーマットに従って個人化されたドキュメントの形にフォーマットされる。

【0019】さらに別の面において、本発明はハイパー・メディア・ドキュメントを処理するためのシステムである。このシステムはハイパー・メディア・ドキュメントにアクセスし、ハイパー・メディア・ドキュメントからアドレスを抜き出し、ハイパー・メディア・ドキュメントから抽出されたアドレスをコンテナに格納する。このシステムは、コンテナに格納されたアドレスで格納されたデータを処理するために、処理ファンクションを起動し、コンテナに格納されたアドレスで格納されたデータを、メモリへダウンロードし、そして予め決定されるデータを、ダウンロードされたデータから、予め決定される形状情報に従って抜き出す。その予め決定されるデータはそれから、フォーマットされたドキュメントを生成させるために、予め定義されるフォーマットの設定に従ってフォーマットされ、そしてこのフォーマットされたドキュメントは処理ファンクションに従って処理される。

【0020】好適な実施形態において、このシステムは、グラフィック・ユーザ・インターフェースを通して、フォーマットの設定および形状情報を入力するものである。このグラフィック・ユーザ・インターフェースは、複数あって、そのうちの1個が処理ファンクションを起動する処理アイコンから成る。グラフィック・ユーザ・インターフェースの能力により、ユーザはドキュメントのフォーマットを対話式に設定することと、そのフォーマットを、必要な場合に変更することができる。特に好適な実施形態において、グラフィック・ユーザ・インターフェースは、複数のモードで表示される。その複数のモードは、(1) グラフィック・ユーザ・インターフェースが、フォーマットする領域、処理オプション、メニューおよび処理アイコンを画面表示するようなフル

機能モード、並びに(2)グラフィック・ユーザ・インターフェースが処理アイコンだけを画面表示するような最小モードから成る。一般的には、最小モードで画面表示されたグラフィック・ユーザ・インターフェースは、ハイパー・メディア・ドキュメントをブラウジングしている間、画面表示される。本発明は、複数のモードでグラフィック・ユーザ・インターフェースを画面表示することにより、ハイパー・メディア・ドキュメントのブラウジングの間、本発明の作業を容易にする。

【0021】この要約は、本発明が速やかに理解されるように提供されるものである。その好適な実施形態についての詳細な説明を、添付の図面と関連させて参照することによって、本発明についてのより完全な理解が得られよう。

【0022】

【発明の実施の形態】第1図は本発明の具体的な実施形態の外観を表わす展望図である。第1図では、マッキントッシュ、或いはIBMパソコン、或いはマイクロソフト・ウインドウズ等のウインドウ画面を有するPC互換コンピュータのような、コンピュータ装置1が説明される。コンピュータ装置1には、カラーモニターかモノクロモニターのようなディスプレイ画面2、テキスト・データおよびユーザ・コマンドを打ち込むためのキーボード3、そして、ディスプレイ2に表示されるオブジェクトを指示したり操作するためのマウス4のようなポインティング手段が設けられている。

【0023】コンピュータ装置1は、また、ディスクドライブ5のような大量記憶手段を含む。画像データは、ネットワーク・インターフェース11aのような種々のソースから、またはファクシミリやモデムのインターフェース6を通じて外部の手段から、コンピュータ装置1に入力することができる。ネットワーク・インターフェース11aは、コンピュータ装置1を、ローカルエリアネットワーク(LAN)に、またはワールド・ワイド・ウェブのようなワイドエリアネットワーク(WAN)に接続するために使用される。

【0024】第2図は、コンピュータ装置1の内部構造を示す詳細なブロック図である。第2図に示されるように、コンピュータ装置1は、コンピュータ・バス9に接続される中央処理ユニット(CPU)8を含む。プリンタ・インターフェース10、ファクス/モデムインターフェース6、ディスプレイ・インターフェース11、ネットワーク・インターフェース11a、キーボード・インターフェース12、マウス・インターフェース13、メイン・メモリ14、ならびにディスクドライブ5もまたそれぞれコンピュータ・バス9に接続される。

【0025】メインメモリ14は、個人用ニュース・プロフィール・エディタ16またはウェブ・プリンタ17のようなアプリケーションを実行するときに、CPU8により使用されるためのランダム・アクセス・メモリ・

格納を提供するように、コンピュータ・バス9とやり取りする。より詳しく言うと、CPU8はこれらソフトウェア・アプリケーションを、ディスクドライブ5から、メイン・メモリ14へロードし、そしてそのソフトウェア・アプリケーションを、メイン・メモリ14上で実行する。ユーザの命令に従って、格納されたアプリケーション・プログラムは起動されデータの処理と操作を行う。典型的には、個人用ニュース・プロフィール・エディタ16、ウェブ・プリンタ17、HTMLフォーマッタ18のようなソフトウェア・アプリケーションは、フロッピーディスクやCDROMのようなコンピュータ読み取り媒体からソフトウェア・アプリケーションをダウンロードすることにより、またはコンピュータ掲示板からソフトウェア・アプリケーションをダウンロードすることにより、ディスクドライブ5に格納されている。

【0026】ディスクドライブ5は、テキスト・ファイルおよび画像ファイルを含むことのできるデータファイルを圧縮フォーマットまたは非圧縮フォーマットで格納し、そして上記のようなソフトウェア・アプリケーション・ファイルを格納する。そのソフトウェア・アプリケーション・ファイルは、ウインドウズ・アプリケーション、DOSアプリケーション、および個人用ニュース検索ファイル15を含む。個人用ニュース検索ファイル15は、個人用ニュース・プロフィール・エディタ16、ウェブ・プリンタ17、HTMLフォーマッタ18、個人用ニュース・プロフィール19およびサイト・プロフィール20を含む。個人用ニュース検索ファイル15の詳細な機能は、個人用ニュース検索システムの作業を簡単に概観した後、説明されよう。

【0027】〈ドキュメント検索の概観〉第3図は第3A図から第3D図までの図から成っており、本発明の具体的な実施形態の実行を図示する。第3A図は、ニュース情報を含む典型的なウェブ・サイト21を図示したものである。ウェブ・サイト21の中にはヘッディング23のようなインデックスへのリンクを有するホームページ22があり、このインデックスは次に記事24にリンクされる。記事24のあるものは他の記事にリンクされる。記事H26はもうひとつのウェブ・サイト上にあるので、リンク25はサイト横断リンクである。リンク25は、ホームページ22に代表される1つのハイパー・メディア・ドキュメントがどのように複数のウェブ・サイトを横断できるのかを説明する。

【0028】ニュースをウェブ・サイト21から検索するために、本発明は、最初にウェブ・サイト21を横断し、ユーザの定義した規則に従ってデータを検索する。これからより詳細に説明されるが、これらの規則はウェブ・サイト21の構造に、またはウェブ・サイト21の構造とその内容に基づくことができる。そのデータは抽出データ・ツリーの形に抜き出され、このツリーは、第3B図に示すようなデータの構成を保持する。尚、同

図において、いくつかのリンクは除外されている。

【0029】抽出データ・ツリー27の構成はいくつかの特徴を有する。その第1として、抽出データ・ツリー27は、サイト29のための1つ以上の子供ノードを有することのできるルート28を有する。このノードは、更に、インデックス／ヘッディング23や記事ノード31などに対応するインデックスノード30を次々に有することのできる。第2に、抽出データ・ツリー27は、いかなるループ（たとえば循環パス）も有しない真のツリーである。たとえば第3A図では、ホームページ22から、インデックス・ノード#1まで、次に記事Cまで、それからホームページ22に戻るループを図示するが、このループは抽出データ・ツリー27を作るときに除去される。

【0030】第2として、抽出データ・ツリー27の構成は、そのウェブ・サイトがどのようにトラバースされるかに依存するのであって、ウェブ・サイトの実際のレイアウトには依存しない。従って、記事H26はインデックス・ノード#3の下に（サイト#1の下に）現れており、このことは、本ニュース検索システムがサイト#1からサイト横断リンク25を通じて記事H26にアクセスしたことを示す。

【0031】最後に、先に言及したことであるが、ある記事は、ウェブ・サイト21の構造のために、あるいはインデックス／ヘッディング23と記事24の内容のために、データ・ツリー27から除外されている。たとえば記事EとGは、抽出データ・ツリー27から除外されている。本発明によると、抽出データ・ツリー27は、もっと多くの除外規則を参照する必要もあるが、第3C図で説明されているような線形ドキュメント32に平坦化される。線形ドキュメント32は、そこに埋め込まれた抽出データ・ツリー27からの情報を有するところの、単純に連続的なドキュメントである。

【0032】最後に、線形ドキュメント32は、ユーザの指定した（即ちデフォルト）フォーマット化指令に従って、第3D図で様式化された個人用新聞として説明される、フォーマットされたドキュメント33の形に定式化される。このフォーマットされたドキュメント33は、サイトのラベル、インデックス／ヘッディング、記事などについて、さまざまなフォントおよび色（またはいずれか一方）を有する。さらに、フォーマットされたドキュメント33はページに分割される。

【0033】本ニュース検索システムの別の実施形態において、上述した、ウェブ・サイト21からフォーマット化ドキュメント33までの変換のある段階をスキップすることができる。たとえば、ウェブ・サイト21からのデータは、同じ記事を二回ダウンロードせずにそしてウェブ・サイト21の構成においてループを避けるために、そのデータの構成のレコードが維持され（あるいは別のリンクされたリストにおいて）る限りにおいて、平

坦化されたドキュメント32の形式に直接に検索することができる。別の実施形態では、抽出データ・ツリー27は、フォーマットされたドキュメント33の形に、直接フォーマットできる。いずれにせよ、本発明の基本的な動作は同じであり、即ち、本ニュース検索システムは、ウェブ上のハイパー・メディア・ドキュメントを横断して、ユーザが定義する情報に従ってデータを抜き出し、そのデータを個人化された新聞の形にフォーマットする。

【0034】上記議論で述べたように、種々のユーザ定義規則および他の情報（フォーマット情報のような）が本ニュース検索の処理に使われる。そのユーザ定義情報は、個人用ニュース・プロファイル19に格納され、その定義は次に説明される。

〈個人用ニュース・プロファイルの定義〉第4図および第5図は、個人用ニュース・プロファイル19が定義される処理を図示する。個人用ニュース・プロファイル19を作るために、個人用ニュース・プロファイル・エディタ16は個人用ニュース・プロファイル19、サイト・プロファイル20およびウェブ・リーダ34と連絡を取る。

【0035】個人用ニュース・プロファイル19は、個人化された新聞を作るためにどのサイトにアクセスすべきか、それらのサイトからどの部門を検索すべきかに関する情報、その部門からどのデータを抜き出すかを決定するために使用される規則、どのようにリンクを除外するかを決定するための規則および新聞フォーマット情報を有する。個人用ニュース・プロファイルの実例が付録1で説明される。

【0036】サイト・プロファイル20は、特定のユーザに特定されない一般的なサイト情報を含む。たとえば、サイト・プロファイル20は、十分なサイト・アドレス、サイト内のセクション、ユーザに特定されないパスワード等のような情報を含むことできる。サイト・プロファイルの実例が付録1で説明される。一般的なサイト情報はサイト・プロファイル20に格納されているので、個人用ニュース・プロファイル19は、サイト・プロファイル20を参照しながら、その一般的なサイト情報を参照し、そのために、個人用ニュース・プロファイルのスペースを節約することができる。たとえば付録1で説明されるように、個人用ニュース・プロファイル19はサイト番号1を参照することができる。サイト・プロファイル20は、サイト番号1が"http://www.sjmercury.com/"にホームページを有する“サンノゼ・マーキュリー・ニュース”であるということを示す。この構造は、また、一般的なサイト情報を中央に集める。従ってサイト・アドレスが変わると、このシステム上の全ての個人用ニュース・プロファイル19を最新ののものにするために、サイト・プロファイル20だけは変更される必要がある。

【0037】ウェブ・リーダ34は、ウェブ・サーバー35を介してWWウェブと通信するアプリケーション・プログラム即ちプログラム・モジュールである。ウェブ・リーダ34は、個人用ニュース・プロフィール・エディタ16からのコマンドに回答してWWウェブにアクセスし、ウェブ上のハイパー・メディア・ドキュメントを横断し、このドキュメントからデータを検索し、そして検索されたデータを個人用ニュース・プロフィール・エディタ16に戻す。

【0038】第4図で説明されたように、個人用ニュース・プロフィール・エディタ16は、サイト・ドライバー36、ウェブ・リーダ・インターフェース37、プロフィール・マネージャ38およびフォーマット・エディタ39の、4つのモジュールを含む。ウェブ・リーダ・インターフェース37は、個人用ニュース・プロフィール・エディタ16をウェブ・リーダ34にインターフェースさせる。サイト・ドライバー36は、ウェブ・リーダ・インターフェース37を通じてウェブ・リーダ34と対話して、個々のウェブ・サイトそれぞれに対してアブストラクト・インターフェースを提供する。より詳細にいうと、サイト・ドライバー36は、ウェブ・リーダ34に指令して、いろいろなウェブ・サイトにアクセスせしめ、それらのサイトからデータを検索させる。その後は、サイト・ドライバー36は、そのデータを受け取り、そこからサイト・プロフィール20を組み立てる。そのデータは、また、現存のサイト・プロフィールを最新のものにするために使用することができる。

【0039】サイト・ドライバー36は、サイト・プロフィール20を組み立てる時に、それぞれアクセスされたウェブ・サイトの構造を、サイト・プロフィール20で定義された均一の構造に変換し、そこから検索されたデータをサイト・プロフィール20に格納する。本発明は、別々のウェブ・サイト（それのあるものは異なる構造を有するかもしれない）をひとつの均一な構造に変換し、そしてウェブ・サイトからのデータをサイト・プロフィール20のその構造に格納することによって、異なるウェブ・サイトからの情報へのアクセスを容易にし、従って全体の処理時間を少なくする。

【0040】プロフィール・マネージャ38は、個人化された新聞をどのようにフォーマットするかを指定するドキュメント・テンプレートを管理する。予め定義されたドキュメント・テンプレートが存在する。さらにフォーマット・エディタ39は、現存のテンプレートを編集するかまたは新しいテンプレートを作るかのどちらによっても、ユーザが、新聞をフォーマットするための個人化されたテンプレートを指定できるようにする。とにかく各々のドキュメント・テンプレートは、個人化された新聞のタイトル、インデックス／ヘディング、副ヘディング、テキスト等についてのページレイアウト情報、フォント情報、スタイル情報、色などを指定する。

【0041】個人用ニュース・プロフィール・エディタ16、サイト・ドライバー36、プロフィール・マネージャ38のためのサンプルのコードが付録3Aに含まれる。第5A図および第5B図は、個人用ニュース・プロフィール・エディタ16の動作をより詳細に説明するフローチャートである。第5A図は、個人用ニュース・プロフィール19のうち、ウェブ・サイトへのアクセスおよびそれらのサイトからのデータ検索に関係する部分を定義するときの、個人用ニュース・プロフィール・エディタ16の動作を説明する。

【0042】第5A図のステップS500において、個人用ニュース・プロフィール・エディタ16がユーザによりスタートされる。ステップS501において、エディタはウェブ・リーダ34を起動させる。ユーザの個人IDがそれからステップS502において検索される。個人用ニュース・プロフィールがそのIDに対して既に存在するならば、ステップS503からステップS504に進み、ユーザがフォーマット・エディタへスキップするという選択を与えられる。そのような個人用ニュース・プロフィールが存在しなければ、個人用ニュース・プロフィール・エディタ16はステップS505の“学習モード”に入る。一旦学習モードに入ると、個人用ニュース・プロフィール・エディタ16はステップS506に進み、そこでユーザからのウェブ・コマンド（即ち、ハイパー・メディア・リンクを横断するというコマンド）を入力し、そのコマンドを、サイト・ドライバー36を使ってウェブ・リーダに転送する。サイト・ドライバー36は、ウェブ・リーダ34が訪れてきたウェブサイトの序列的なログを維持する。ステップS507では、個人用ニュース・プロフィール・エディタ16は、そのウェブ・コマンドから抽出規則を創成する。この規則により、本ニュース検索システムは、ユーザがウェブサイトをブラウジング（内部のハイパーリンク上をクリックする）する際における選択基準を後で繰り返すことを可能にする。

【0043】その規則は、少なくとも、ウェブサイトの横断を繰り返すための構造的な基準を指定する。たとえば、ユーザが特定のインデックス／ヘディングの下にある記事すべてにアクセスするならば、この規則は、そのインデックス／ヘディングの下全ての記事は検索されるものであると指定するはずである。本発明のひとつの実施形態では、上記規則は、ユーザから入力された、内容に基づく（content-based）基準（即ち、キーワードに基づく基準）を含むこともできる。これら内容に基づく規則は、たとえば、（1）ある単語がひとつの記事に在るべきことを要求することができる、（2）ある語を有する記事を除外することができる、（3）単語のブル組み合わせを要求することができ、（4）キーボードに基づく順位を有し構造上の基準に基づいて選択される記事をランク付けして、それから最高順位を有する

記事の選択を要求することができる、または（５）広告のようなあるタイプの記事を除外することができる。

【0044】構造的でそして内容に基づくcontent-based除外規則のためのシンタックスの例が付録２で説明される。異なる型の規則がいくつか示される。ウェブサイトの横断をある数のリンクに限定するだけの規則もある。また他のものは、日付およびキーワードに基づいた除外ルールもある。ひとつの特に順応性のある規則は、記事がキーワード分析に基づいてランク付けされ、最高点の記事が選択されるようにするものである。他のルールは“平坦化”規則を含む。これらの規則は、後に詳細に説明されようが、抽出データ・ツリーの平坦化を制御する。

【0045】少なくとも、この規則は、ユーザの選択（即ち、第１ページ、第１のドキュメント、全てのリンクなど）、必要なパスワード情報、ブラウザーコマンドなどについての構造上の情報を含む。この規則は、また、サイト・プロファイル２０へのポインタ即ち参照、並びに、その中の適切な情報を含む。サイト・プロファイル２０を保守するために、一般的な（ユーザが指定しない）情報がサイト・ドライバー３６により使用される。この方式では、先に論じらアドレス情報と複数のユーザに共通のパスワードとが、上述したように、サイト・プロファイル２０において保守することができる。たとえば、サイト・ドライバー３６は、コマンドまたは他のドキュメントへのハイパーリンクを、この規則内のウェブページに格納するが、ウェブサイトの完全なアドレスをこの規則に格納することはない。そのアドレス情報はサイト・プロファイル２０に格納される。

【0046】ステップＳ５０８では、ウェブ・コマンドから作られた規則を定義する規則データは、第３Ｂ図のような抽出(extracted)データ・ツリー２７のような抽出データ・ツリーに格納される。このデータ・ツリーは、ウェブから検索されるデータの構成を反映するリンクのリストである。ステップＳ５０９では、ユーザが終了して（即ち、ユーザがウェブサイトを終了して）いなければ、フローは、次のウェブ・コマンドを求めてステップＳ５０６に戻り、終了していれば、フローはステップＳ５１０へ進む。

【0047】この点で、個人用ニュース・プロファイルの創成は、サイト・プロファイル２０が格納要求を最小にし、そして一般的なサイト情報を集中化させるために使用されて来た点を除いて、ワード処理プログラムに共通するマクロ命令の創成に大いに類似してなされてきた。格納要求をさらに最小にするために、そしてニュース検索システムの順応性をより高めて効率のよいものにするために、抽出規則はここで編集され、余分のリンクや、同じサイトへの複数のリンクアクセスなどを取り除く。このことはステップＳ５１０でなされ、結果としてまとめられた規則は、個人用ニュース・プロファイル１

９の最初の部分となる。

【0048】別の形態では、個人用ニュース・プロファイル・エディタ１６は、前もって格納された個人用ニュース・プロファイルをユーザが編集できるようにするか、または、たとえばニュースサイトやヘッディング記事だけやキーワードなどを指定することにより、ユーザがドキュメント構成のプレファレンスを指定できるようにするグラフィカル・ユーザ・インタフェースとして呼び出されても良い。どちらの場合においても、その結果は個人用ニュース・プロファイル１９であり、ウェブサイトへのポインタの一覧と、１つ又は複数のウェブサイトを横断するための抽出規則からなる。

【0049】上記の説明をよりよく理解するために、個人用ニュース・プロファイルの実施形態およびサイト・プロファイルの実施形態が付録１に示される。次に、第５Ｂ図で説明されるように、制御は進んで、ユーザに、カスタムの新聞用テンプレートを修正するための選択を与える。即ち、ステップＳ５１１で、新聞用テンプレートが定義されて、個人用ニュース・プロファイル１９に格納されたか否かが決定される。もし新聞用テンプレートが定義されているのならば、ステップＳ５１２において、ユーザは、テンプレートを編集するか、またはステップＳ５２０へ進行するかを選択を与えられる。ユーザがテンプレートの編集を選択する場合、またはどの新聞用テンプレートも定義されていなかった場合には、フローはステップＳ５１３へ進む。

【0050】ステップＳ５１３では、ユーザは、カスタム・テンプレートを作るか、または予め定義されたテンプレートを使用するかを選択を行う。ユーザが予め定義されたテンプレートを使用したい場合は、ステップＳ５１４で、指定した予め定義されたテンプレートを得、ステップＳ５１９において、それを個人用ニュース・プロファイルに加える。そうしたくない場合は、フローはステップＳ５１５へ進み、そこではフォーマットエディタ３９が呼び出される。

【0051】フォーマットエディタ３９は、ユーザに数多くのフォーマット用オプションを提供するグラフィック・ユーザ・インターフェースを有する。ステップＳ５１６では、フォーマットエディタ３９は、どの新聞のセクションを本新聞に印刷すべきか、どのウェブサイトのニュース記事をそれぞれのセクションに置くべきか、それぞれの欄をどのようにレイアウトすべきかの、いずれかまたはどちらもユーザが定義できるようにする。この点で、どのウェブサイトのニュース記事を第１面として使用すべきか、どのウェブサイトのニュース記事をビジネス欄として使用すべきか、どのウェブサイトのニュース記事をスポーツ欄として使用すべきかなどを、ユーザは指定することができる。さらにステップＳ５１６で、どのような副ヘッディングがそれぞれの欄に合うかだけでなく、それぞれのインデックス／ヘッディングをどこ

にリストすべきか、ユーザは定義することができる。

【0052】ステップS517では、フォーマットエディタ39は、ニュース記事の、インデックス／ヘディング、副ヘディング、筆者名を記す行（バイライン）、および本文のためのフォント形式を、ユーザが定義できるようにする。ステップS518では、フォーマットエディタ39は、ユーザに、インデックス／ヘディングの色、タイトル色などを定義させる。この点では、レイアウトエディタ39は、ユーザに利用できるフォントおよび色の型を、このシステムのプリンタ能力に

基づいて決定することができる。

【0053】一旦全ての情報がカスタム・テンプレートのために集められると、ステップS519で、フォーマットエディタはその情報を個人用ニュース・プロファイル19に加える。変形例では、プロファイルマネージャー38もまた、このカスタムフォーマットをテンプレートとして、他のユーザに使用させることができる共通の領域に格納可能できるようにしてもよい。この場合、そのカスタム・テンプレートへのポインター若しくは参照だけを個人用ニュース・プロファイル19に格納する。

【0054】ステップS520では、個人用ニュース・プロファイル・エディタ16は、自動的な新聞配送の時間および方法をユーザに設定させる（即ち、印刷するか、後で印刷するためにディスクドライブ5に格納する）。これらの設定は個人用ニュース・プロファイル19に加えられる。より具体的にいうと、ユーザのコンピュータに電気が提供され続けている場合には、本ウェブニュース検索システムは指定された時間に自動的に起動される。本システムは、個人用ニュース・プロファイル19にリストされているウェブサイトから記事を検索することになる。ニュース記事を検索すると、その記事は、個人用ニュース・プロファイル19にある新聞テンプレートに基づいてフォーマットされる。フォーマットされた個人化された新聞はそれから印刷されるか、後で見るために格納される。新聞配送の時間が設定されない場合は、ユーザはいつでもこのウェブニュース検索システムを実行することができる。

【0055】個人用ニュース・プロファイル19が一度作られてしまうと、このウェブニュース検索システムは起動されると、ウェブニュース・サイトを横断して、いろいろのニュース記事をウェブニュース・サイトから自動的に検索することによって、個人化された新聞を組み立て、そして個人用ニュース・プロファイル19に示された新聞用テンプレートに基づいて、ニュース記事を印刷することができる。本発明のウェブニュース検索システムがどのようにしてこの機能を実行するかについては次に説明される。

【0056】〈個人用ニュース・プロファイルを用いたドキュメントの検索〉第6図は、本発明が個人用ニュース・プロファイル19に従って、ウェブから記事を検索

する方式に関する具体的なブロック線図である。（第6図は、また、検索された記事が線形のドキュメントの中に平坦化されてフォーマットされる方式も説明する。これらの機能は、この明細書の次の項においてより詳細に説明される。）

第6図で示されるが、ウェブ・プリンタ17はニュース記事の検索を引き受ける。ウェブ・プリンタ17は最終のユーザ用アプリケーションであり、本発明の機能を実行するために、個人用ニュース・プロファイル19、サイトプロファイル20、ウェブ・リーダー34、および出力インターフェース40と連絡する。

【0057】ウェブ・プリンタ17は、どのウェブ・サイトにアクセスすべきか、これらのサイトからどのデータを検索すべきかを決定するために、個人用ニュース・プロファイル19を見る。ウェブ・プリンタ17はまた、一般的なサイト情報を求めてサイト・プロファイル20も見る。ウェブ・プリンタ17は、種々のウェブ・サイトにアクセスし、そのサイトからデータを検索する目的で、個人用ニュース・プロファイル19並びにサイト・プロファイル20にある情報に従って、ウェブサーバー35を通じてウェブに接続するようにウェブ・リーダー34に指令する。ウェブ・リーダー34は検索されたデータをウェブ・プリンタ17に送る。ウェブ・プリンタ17はそのデータを使用して抽出データ・ツリーを組み立てる。ウェブ・プリンタ17は、それから、抽出データ・ツリーを平坦化して線形ドキュメントに変換し、そして、その線形ドキュメントを出力インターフェース40を通じて出力用にフォーマットする（尚、この明細書の次の項においてより詳細に説明される）。

【0058】第6図で説明されるように、ウェブ・プリンタ17は、ウェブ・リーダー・インターフェース50、サイトドライバー51、ツリー・マネージャー41、およびフォーマッタ42という、4つのプログラム モジュールを含む。ウェブ・リーダー・インターフェース50は、上記のウェブ・リーダー・インターフェース37のように、ウェブ・プリンタ17をウェブ・リーダー34に接続する。

【0059】サイトドライバー51は、サイトプロファイル20および個人用ニュース・プロファイル19にアクセスし、そこに格納されたデータをウェブ・リーダー34に提供する。上記のように、ウェブ・リーダー34はそのデータを使って、いろいろのウェブ・サイトにアクセスし、そこからデータを抽出する。上記のように、この取り出されたデータは、ウェブ・プリンタ37により、抽出データ・ツリーを組み立てるために使用される。

【0060】ツリー・マネージャー41は抽出データ・ツリーを管理する。この点で、ツリー・マネージャー41は取り出されたデータの構成の前記抽出データ・ツリーにおける経路をキープする。これにより、ウェブ・プリンタ17が、同じ記事に二度アクセスすることが防止

され、ウェブサイト unnecessarily 再訪することが防止され、また、ウェブ上のあるハイパー・メディア・ドキュメントの組織にあるサイクル（ループ）に捕まることを避けることができる。別の形態では、ツリー・マネージャー41は、抽出データ・ツリーと同じ機能性を提供するリンクされたリストを参照して、そのデータをブロック（データ・ツリーの中に直接格納する方式とは反対に）に格納することができる。ツリー・マネージャー41のためのサンプルコードは付録3Bに含まれる。

【0061】フォーマット42は、抽出データ・ツリーを平坦化して線形ドキュメントにすること、およびその線形ドキュメントを個人化された新聞の形にフォーマットすることを引き受ける。フォーマット42は、その印刷基準と、個人ニュース・プロファイル19に示されたフォーマット情報（即ち、新聞テンプレート）に従ってそれらの機能を実行する。フォーマット42のためのコードの見本は付録3Bに含まれる。

【0062】より詳細に言うと第7図は、流れ図である。ウェブ・プリンタ17が、ウェブ・リーダ34を使って、どのようにして、個人用ニュース・プロファイル19に従ってウェブを横断し、そのプロファイルに従って、どのようにして、望まないデータを除外しながらウェブから記事を取り出すかを説明するフローチャートである。

【0063】ウェブ・プリンタはステップS700から開始する。ステップS701で、ウェブ・プリンタ17は、ユーザの指定した個人用ニュース・プロファイルか、または、サイトドライバ51を使用するディスクドライブ5に格納されたデフォルト個人用ニュース・プロファイルのどちらかを検索する。この点で、コンピューター装置1は一人以上のユーザにより使用される可能性があるために、1個かそれ以上（そのうちの1個はデフォルトとして指定される）個人用ニュース・プロファイルが装置に格納されてもよい。その指定された個人用ニュース・プロファイルを取り出すとすぐに、ウェブ・プリンタ17は、ステップS702で、なんらかのニュースデータが前にディスクドライブ5（即ち、前になされた自動ニュース送付によって）に格納されているか、またはニュース記事が個人用ニュース・プロファイル19を使用して検索されるべきかを決定する。

【0064】ニュースデータがディスクドライブ5に存在する場合には、ステップS703で、その格納されたニュースデータが取り出され、フローは、次の項で詳述される第8図のステップS801に進行する。これに対し格納されたニュースデータが何も存在しない場合、ウェブ・プリンタ17はステップS704でウェブ・リーダ34を呼び出す。この動作は、個人用ニュース・プロファイルの定義に関して前に説明されたものと同じウェブ・リーダ34であることに留意されたい。

【0065】呼び出されると、ウェブ・リーダ34は、

ステップS705で、ワールド・ワイド・ウェブのようなネットワークへの接続を提供するウェブサーバー35に接続する。それから、ウェブ・プリンタ17は、個人用ニュース・プロファイル19から検索された情報に基づいて、訪ねるべき一番目のウェブサイトのためのアドレスをウェブ・リーダ34に提供する。ステップS706で一度目的のウェブサイトに接続されると、ウェブ・プリンタ17は、次のウェブページへウェブを横断して行くためのコマンド／リンクをウェブ・リーダ34に提供する。このコマンド／リンクは、個人用ニュース・プロファイル19が検索すべきであると示す情報を含む。ステップS707で、ウェブ・リーダ34はこの情報に従ってウェブを横断する。

【0066】ステップS708で、ウェブ・リーダ34は、個人用ニュース・プロファイル19にある規則に従って目的の情報を取りだし、それをウェブ・プリンタ17に送る。従って、データの除外はこのステップにおいて行われる。個人用ニュース・プロファイル19の規則は、個人化された新聞からデータを除外するための、構造的な内容に基づくcontent-based基準を指定する。この構造的な規則は、ウェブ・リーダ34によりアクセスされたウェブサイトの構造に基づき、その取り出された情報を制限する。前記内容に基づく基準は、その内容に基づいてその取り出された情報を制限する。個人用ニュース・プロファイルを作成する方法に関してすでに説明されたように、個人用ニュース・プロファイル19の取り出し規則のシンタクスの例は付録2に含まれる。

【0067】規則に基づいた除外に加えて、ステップS708ではメディアの型式による除外が行われる。即ち、印刷不可能なメディア型式のデータが抽出データ・ツリーから除外される。たとえば映画および音響データが除外できる。ウェブ・プリンタ17は、取り出したデータを、ディスクドライブ5（またはメインメモリ14）の、ツリーマネージャー41により管理される抽出データ・ツリーに格納する。別の形態では、上記データは、すでに説明されたように、リンクされたリストを参照してブロックに格納することができる。ウェブ・プリンタ17は、ステップS709で、ウェブサイトのウェブページから全ての情報を完全に取り出すために、ステップS707へ戻る。ひとつのウェブサイトを完全に検索すると、ステップS710で、ウェブ・プリンタ17はさらに多くのサイトを訪ねるべきかどうかを決定するために、ツリーマネージャー41を使用して、個人用ニュース・プロファイル19に残るサイトを、抽出データ・ツリーにあるサイト構成情報と比較する。さらに多くのウェブサイトを訪ねる必要がある場合は、ステップS710からステップS706に戻り、ここでニュース記事は上記と同じ方式で取り出される。これに対して、個人用ニュース・プロファイル19に記録されているウェブサイトのすべてが訪ねられて、記事がすべて取り出さ

れてしまうと、フローは第8図のステップS801に進む。

【0068】〈取り出しデータの平坦化とフォーマット〉第8図は抽出データ・ツリーがどのように平坦化(flattened)されてフォーマットされるのかを示すフローチャートである。この発明の構成は、ウェブからデータを検索するとき(第6図で説明される)と同じである。実際、平坦化処理およびフォーマット処理は、少なくともある限度までにおいて、データ検索取り出し処理と同時に行うことができる。

【0069】第8図のステップS801で、抽出データ・ツリーは平坦化される。つまりデータの構造が抽出データ・ツリーから線形ドキュメントへ転換されるということである。この行程は個人化された新聞からより多くのデータを除外する機会を提供するもので、たとえば前記データ・ツリーのノードを平坦化されたドキュメントの中へ含めるだけでよい。この除外処理は個人用ニュース・プロファイル19にある平坦化規則により制御される。

【0070】データが線形ドキュメントの形に平坦化された後、そのデータはステップS802で、個人用ニュース・プロファイル19で示されるテンプレートに従ってフォーマットされる。このテンプレートの定義は、予め定義されたテンプレート、またはカスタムのテンプレート、のどちらの場合もすでに論じられている。最後にフォーマットされて完全に個人化された新聞は出力インターフェース40に送られる。このインターフェースは、プリンタ7に接続するプリンタインターフェース10、ディスプレイ2へ接続するディスプレイインターフェース11、またはモデム/ファックス・インターフェース6でも可能である。

【0071】〈第2の実施形態〉… HTMLフォーマッタ本発明の第2の実施形態はハイパー・メディア・ドキュメントを処理するためのシステムである。このシステムはハイパー・メディア・ドキュメントにアクセスし、ハイパー・メディア・ドキュメントからアドレスを抽出し、そしてハイパー・メディア・ドキュメントから抽出されたアドレスをコンテナに格納する。このシステムは、コンテナに格納されたアドレスの番地に格納されたデータを処理するという処理機能を起動し、コンテナに格納されるアドレスに格納されるデータをメモリにダウンロードし、そして予め決定された構成の情報に従って、ダウンロードされたデータから予め決定されたデータを抽出す。その予め決定されたデータはそれから、フォーマットされたドキュメントを生成するための、予め定義されたフォーマットセッティングに従って、フォーマットされ、そしてフォーマットされたドキュメントは処理ファンクションに従って処理される。本発明の第2の実施形態は、第2図で言及されたHTMLフォーマッタ18として説明される。

【0072】HTMLフォーマッタ18の例は、本出願人のCanon Information System Incにより製造されたウェブ・フォーマッタである。第2の実施形態はウェブ・フォーマッタに関して説明される。しかしながらHTMLフォーマッタ18はウェブ・フォーマッタの実施形態に限定されないこと、そして以下の説明の精神および範囲の内に異なる別々の様々な実施形態が可能であることに留意されたい。

【0073】ウェブ・フォーマッタは、ネットスケープ、モザイクおよびインターネット・エクスプローラのようなウェブ・ブラウザと共に使用できる独立型のユーティリティ・ソフトウェアである。要するに、ウェブ・フォーマッタはウェブページからデータを抽出し、その抽出されたデータから、突発的に発生したデータを除去し、そしてその除去後のデータをフォーマットされたドキュメントの形に再フォーマットする。フォーマットされたドキュメントは印刷されるか、RTF(Rich Text Format)形式で格納されるか、またはマイクロソフト・ワード(MS Word)、ワード・パーフェクト(Wordperfect)、ワード・パッド(Wordpad)等の、あらゆるRTFコンパティブル・エディタにおいて編集されることができる。ウェブ・フォーマッタは、MicrosoftのWindows(登録商標)のようなウインドウ環境から起動することができる。そのようなウインドウ環境から、ウェブ・フォーマッタは、起動ウインドウ内のウェブ・フォーマッタ・アイコン(不図示)をダブルクリックし、Windowsのスタートメニューからウェブ・フォーマッタを選択し、所定のウェブ・ブラウザからURL(uniform resource locator)アイコン(不図示)をドラッグし、そしてそれをウェブ・フォーマッタ・アイコンの中へドロップすることにより、もしくはウェブ・ブラウザが開始されたときにウェブ・フォーマッタを自動的に呼び出すことにより、起動することができる。

【0074】上記第1の実施形態とは異なって、ウェブ・フォーマッタは、1個または2個以上のウェブのページから特定の型式のドキュメントを創成するための基準を指定するために、予め定義された個人用ニュース・プロファイルを使用することはない。むしろ、第2実施形態のウェブ・フォーマッタは、1個または2個以上のウェブのページから新聞等のような特定の型式のドキュメントを創成するために、ユーザ指定の基準に依拠する。これらの基準はユーザにより、グラフィック・ユーザ・インターフェースを通じて対話式に入力される。

【0075】以下でより詳細に説明されるが、ウェブ・フォーマッタは二つのモード即ち最小機能モードおよびフル機能モードで作業する。最小機能モードでは、ウェブ・フォーマッタのグラフィック・ユーザ・インターフェースは、必ず、画面に表示されるウェブのページと同時に表示されるフローティング・プリント・ボタンでなければならない。この特徴によって、ユーザがウェブを

探索するときに、フローティング・プリント・ボタンをクリックするだけで、ユーザはウェブのページを処理すること、フォーマットすること、およびプリントアウトすることができる。

【0076】ウェブ・フォーマッタのフル機能モードでは、ウェブ・フォーマッタのグラフィック・ユーザ・インターフェースは、ユーザに対して、処理対象のウェブのページのURLアドレスを入力するためのインターフェース・スペースと、当該ドキュメントの個人用のタイトルを入力するためのインターフェース・スペースと、そのドキュメント用の形式（フォーマット）を選択するためのインターフェース・スペースと、フォーマットされたドキュメントの第1ページをプレビュー(preview)するためのインターフェース・スペースと、そして、ドキュメントを印刷するか、またはドキュメントをRTFファイルとしてセーブするか、またはRTFエディタを使用してそのドキュメントを見る／編集するかするためのインターフェース・スペースを提供する。まず、フル機能モードのためのグラフィック・ユーザ・インターフェースから最初に説明される。なぜなら、ユーザが最小機能モードを選択できるのはフル機能モードのインターフェースからであるからである。

【0077】第9A図はウェブ・フォーマッタのフル機能モードのためのグラフィック・ユーザ・インターフェース43を説明する。このグラフィック・ユーザ・インターフェース43は、ウェブ・フォーマッタの最初の起動後すぐに画面に表示される。あらゆる対話式のウインドウ形式のソフトウエア・アプリケーションの場合と同様に、ユーザはグラフィック・ユーザ・インターフェース43と、マウス4およびキーボード3によって（指示し、クリックすることにより）対話する。

【0078】第9A図で示されるように、グラフィック・ユーザ・インターフェース43は、その初めから終わりまでの間で、ユーザがフォーマットすべきドキュメントのURLアドレス、およびそのドキュメントのフォーマットを指定できる、フィールド44および46から49までを含む。フィールド44から始めると、ユーザは、ウェブ・フォーマッタにより処理すべきウェブのページのURLアドレス（e.g., http://www.cis.canon.com/tis/tis home.htm）を入力する。ユーザがURLアドレスを入力するにはいくつかの方法がある。ユーザは、（1）アドレスをURLフィールド44へ直接にタイプするか、（2）ウェブ・ブラウザでURLアドレスをコピーして、そのURLアドレスをURLフィールド44に貼り込むか、（3）URLアドレスをウェブ・ブラウザから、グラフィック・ユーザ・インターフェース43へ、またはウェブ・フォーマッタのアイコンへドラッグするか、または（4）カレントURLボタン54をクリックする。

【0079】カレントURLボタン54に関して、ユーザがカレントURLボタン54をクリックすると、ウェブ・

フォーマッタはアクティブなウェブ・ブラウザを捜し当て、ウェブ・ブラウザに、現在のウェブのページのアドレスに関して訪ねる。次にウェブ・ブラウザは現在のウェブのページのアドレスをウェブ・フォーマッタに対して提供し、ウェブ・フォーマッタはそのアドレスをURLアドレス・フィールド44に置く。URLボタン(Current URL)54が起動され、どのウェブブラウザも目下ランしていない時は、ウェブ・フォーマッタは、第9A図の・ダイアログ・ボックス56を表示する。

【0080】同図に示すように、・ダイアログ・ボックス56は、キャンセル(Cancel)ボタン57およびブラウザ・スタート(Launch Browser)ボタン59を含む。キャンセルボタン57は、URLアドレスをCurrent URLボタン54を介してURLアドレスフィールド44に入力するようにというユーザの要求をキャンセルするものである。これに対してブラウザ・スタート・ボタン59は、ウェブ・フォーマッタで指定されるウェブ・ブラウザを開始する。以下で言及されるが、ウェブ・フォーマッタは、当該ウェブ・フォーマッタと共に使用されるウェブ・ブラウザを含む所定の情報により前もって構成立てられている。ウェブ・フォーマッタの構成は以下でより詳細に説明されよう。

【0081】ウェブ・フォーマッタの別の実施形態では、ファイル名をURLアドレス・フィールド44に入力してもよい。たとえばこれら選択可能な実施形態は、ユーザがハイパー・リンクされたマニュアルを書籍形式のフォーマットにフォーマットしたい時には、ユーザはファイル名をURLアドレスフィールド44に入力する。その後ウェブ・フォーマッタは、指定されたウェブのページを介する時と同様の方式でそのファイル进行处理して、そのハイパー・リンクされたマニュアルを望ましいように再フォーマットする。

【0082】グラフィック・ユーザ・インターフェース43に論を戻すと、タイトル・フィールド(Title)46は、ユーザがフォーマットされたドキュメントのための、個人用タイトルを入力できるようにする。そのタイトルは直接タイプして、タイトルフィールド46に貼り付けることも可能である。フォーマット用フィールド(Styles, Columns, Spacing)47乃至49は、ウェブ・フォーマッタによって出力されるべきドキュメントのフォーマットを定義する。他のフォーマット用フィールドを選択するには、スクロールバー55のような、それぞれのフォーマットフィールド選択用のスクロールバーをクリックすることによりアクセスされる。これらのフィールドの各々は以下で詳細に説明される。

【0083】スタイル(Styles)フィールド47は出力ドキュメントをフォーマットするための4個のオプションを提供する。これらのスタイルは、小ヘッディング(headers)、余白(margins)等の大きさの如き、出力ドキュメ

ントの特性に関するものである。スタイル・オプションは、コンテンポラリー(contemporary)、フォーマル(Format)、ファン(Fun)およびプロフェッショナル(Professional)を含む。もちろん本発明はこの4個のスタイルのオプションに限定されず、必要なら他のスタイルを加えることができる。

【0084】コラム(Column)フィールド48はフォーマットされた出力ドキュメントのコラムの数を定義する。2つのコラムのオプションが利用でき、即ち単一コラム(Single)および多重コラム(Multiple)である。しかしながら本発明はこの2個のオプションに限定されない。単一コラムオプションは、要望があればドキュメントをシングル・コラムの形にフォーマットする。これに対し、多重コラムオプションは、ドキュメントを、予め指定された数のコラムの形にフォーマットする。本発明の好適な実施形態では、マルチプルオプションは2個のコラムに対して設定される。しかしながらいかなる数でも設定できる。

【0085】スペーシング(Spacing)フィールド49はフォーマットされた出力ドキュメントの行間を定義する。3個のオプションがウェブ・フォーマットにおいて提供されているが、しかし他のオプションも要望通りに加えることができる。これら3個のオプションは、圧縮(Condensed)、ノーマル(Normal)、および読み易い間隔(Easy-to-read)であって、そのうち圧縮は最小の行間、読み易い間隔は最大の行間である。

【0086】グラフィック・ユーザ・インターフェース43はまたプレビュー(Preview)ボタン60を提供する。プレビュー・ボタン60をクリックすることによって、ユーザはビューイングエリア61で、フォーマットされたドキュメントの最初のページを試しに見ることができる。試し見のためのフォーマットされたドキュメントの一例が第9A図で示される。

【0087】第9Aおよび9B図で示されるように、ウェブ・フォーマットはまたオプション(Options)ボタン61を含む。オプション・ボタン61は、フォーマット用オプションをさらにユーザに提供するもので、ウェブ・フォーマットにより使用されてフォーマットされたドキュメントを創成する。ユーザはオプションボタン61をクリックすることによって、アクティブにすることができる。これにより第9B図のオプション・ダイアログ・ボックス62がディスプレイ2に現れる。

【0088】第9B図で示されるように、オプション・ダイアログ・ボックス62は、一般(General)オプション64、コンテナ(Container)オプション66、およびストリップ・メタ情報(Strip Meta Info)オプション67を含む。「一般」オプション64は「テキストのみ(Text Only)」というリストボックス72、「ページ内リンクのインデックス(Index of links in the page)」というリストボックス73、および、「浮動画無し(No fl

loating pictures)」というリストボックス74を含む。これらのオプションは、各々のリストボックスにおいてチェックマークなどにより選択されるように指示される。オプションについての説明からも明らかであるが、「一般」オプション64のオプションの2個以上を同時に選択することができる。

【0089】「テキストのみ(Text Only)」リストボックス72は、ウェブページにある画像のすべてを取り除いて、そこにあるテキストだけを印刷するように、ウェブ・フォーマットに命令する。「ページ内リンクのインデックス(Index of links in the page)」リストボックス73は、ウェブ・フォーマットに命令して、ウェブページに現在ある全てのURLのリストをフォーマットされたドキュメントの末尾に加えさせる。好適には、URLのリストは上付き文字として印刷され、リストにあるURLの末尾の位置は肉太活字で示される。「浮動画無し(No floating pictures)」リストボックス74は、ウェブ・フォーマットに命令して、フォーマットされたドキュメントの特定の領域でドキュメントの全ての画像を印刷させる。従ってこのオプションが選択されると、画像が特定の領域にぴったりとはまるように、ウェブ・フォーマットは画像を必要なだけ縮小させる。

【0090】ストリップ・メタ情報(Strip Meta Info)オプション67は、ウェブ・フォーマットにより処理されるウェブのページから不必要な情報を容易に除去させるエンジニアリング用のオプションである。このオプションは、

- (1)「無し(None)」：ウェブのページから何も除去しないようにウェブ・フォーマットに命令する、
- (2)「第1の水平ルールまで(Till the first horizontal rule)」：予め定義された第1および第2の水平のフォーマット規則に至るまで(たとえばページを横断する水平の線に至るまで)、全てのリンクと画像を取り除くように、ウェブ・フォーマットに命令する、
- (3)「第1のテキストまで(Till the first text)」：ウェブのページで最初および最後のテキストが発生するまで、リンクおよび画像をすべて取り除くように、ウェブ・フォーマットに命令する。

【0091】ストリップ・メタ情報(Strip Meta Info)オプション67は一度に1つだけ選択することができる。第9B図で示されるように、そこでの選択は、そのオプションに隣接する小さな丸内に点を付すことによって指示される。コンテナ(container)オプション66は、第9B図に示されているように、ドキュメントと、そのドキュメントのコンテナ76に格納アドレスを処理するためのオプションを提供する。コンテナオプション66の説明に先立ち、コンテナ76が説明される。

【0092】コンテナ76は選択されたドキュメントのURLアドレスを格納する。フィールド44に入力されるドキュメントのアドレスはコンテナ76に追加される。

URLがコンテナ76に入力される順序は、URLのデータがウェブ・フォーマットにより処理される順序を意味する。第9B図で示されるように、コンテナ76は一度フルになると、そのアイコンは番号77により示されるアイコンに変化する。

【0093】ユーザがコンテナ76のアイコンをクリックすると、メニュー77が画面に表示される。メニュー77は5個のオプション、即ち、「開く(Open)」79、「空にする(empty)」80、「プリントする(Print)」81、「編集する(Edit)」82および「記憶する(Save)」84を提供する。これらのオプションはアクティブにされると、ハイライトが点くが、詳細は以下で説明される。

【0094】「開く(Open)」79はアクティブにされると、第9B図のコンテナ・コンテンツ(Container Contents)スクリーン87を画面に表示する。コンテナ・コンテンツ(Container Contents)スクリーン87は、4個のボタン、即ち、現在のURLをコンテナ76に加える「URL追加(Add Current URL)」ボタン88と、ユーザがコンテナ76にあるURLをハイライトさせて除去できるようにする「デリート>Delete)」ボタン89と、ユーザがコンテナ76を空にできるようにする空(Empty)ボタン90、およびユーザがコンテナコンテンツ・スクリーン87を閉じることができるようにする終了(Done)ボタン91を提供する。忘れてならないのは、ユーザはまた、メニュー77の「空にする(empty)」80をクリックすることにより、コンテナ76の内容を空にすることもできるということである。

【0095】さらにコンテナ76に格納されたURLの順序を、異なるURL同士をドラッグしドロップすることにより、変更することもできる。先に言及したが、URLはコンテナ76に現れる順序に従って処理されるため、上述の特徴により、ユーザはコンテナ76にあるURLの処理順序をそれぞれ変更することができる。プリント(Print)ボタン81、編集(Edit)ボタン82およびセーブ(Save)ボタン84は、アクティブされた時、コンテナ76に格納されるURLにより定義されるウェブのページで全てのデータを、ウェブ・フォーマットにダウンロードさせ、それらのデータをユーザの指定通りにフォーマットさせ、フォーマットされたウェブのページを格納するRTFファイルを創成させ、そしてセーブするか、編集するか、またはRTFファイルを印刷するかを選択されたアクションを行わせる。この処理は以下でより詳細に説明される。

【0096】オプション・ダイアログ・ボックス62に論を戻すと、コンテナオプション66は、「内容をテーブル印刷(Print table of contents)」リストボックス92と「処理後に空にする(Empty after processing)」リストボックス94を含む。説明のように、リストボックスのチェックマークは、リストボックスが選択された

ということを示すために使われる。この点で、2個以上のリストボックスが一度に選択されることができる。

「内容をテーブル印刷(Print table of contents)」リストボックス92が選択されると、コンテナ76にある全てのURLのタイトルをフォーマットされた出力ドキュメントの内容リストとして印刷するように、ウェブ・フォーマットに命令する。「処理後に空にする(Empty after processing)」リストボックス94がアクティブになると、ウェブ・フォーマットに命令して、印刷か、編集か、またはドキュメントのセーブが済んだ後に、ユーザがそうするのを待たずとも、自動的にコンテナ76を空にさせる。

【0097】コンテナオプション62の一部として、「RTFエディタの選択>Select RTF Editor)」ボタン69、キャンセル(Cancel)ボタン70、およびオーケー(O)ボタン71もまた設けられている。「RTFエディタの選択>Select RTF Editor)」ボタン69をクリックすることによって、ユーザはRTFファイル・エディタ(その例は上述した)を選択することができる。この選択は、たとえば、所定のRTFエディタ(不図示)を一覧表示するもうひとつのダイアログ・ボックスを表示し、そしてそのRTFエディタのうちの1個を選択することによって、なされる。キャンセルボタン70はコンテナオプション62をキャンセルし、オーケーボタン71はコンテナオプション62で選択されたオプションを確認し、それからそのダイアログ・ボックスを閉じる。

【0098】第9A図で示されているが、グラフィック・ユーザ・インターフェース43は、また「印刷(Print)」アイコン96、「編集(Edit)」アイコン97、「セーブ(Save)」アイコン99、「ヘルプ(Help)」ボタン100、「終了(Done)」ボタン101、および「最小化(Minimizing)」アイコン102を含む。ユーザはマウスを使用してそれらをクリックすることにより、それらの動作のどれでも選択することができる。

【0099】「印刷(Print)」アイコン96は、ウェブ・フォーマットによりフォーマットされるウェブのページのコピーを、ユーザが何枚でも印刷できるようにするプリント・ダイアログ・ボックス(不図示)を開く。

「編集(Edit)」アイコン97は、フォーマットされたウェブのページを格納するRTFファイルを開き、それを予め決められたRTFエディタによって編集可能にする。

「セーブ(Save)」アイコン99は、ユーザがフォーマットされたウェブのページに名前を付け、RTFファイルとしてセーブできるようにするセーブ・ダイアログ・ボックス(不図示)を開く。「ヘルプ(Help)」ボタン100は、作業しているウェブ・フォーマットのためのヘルプメッセージを提供し、そして「終了(Done)」ボタン101はウェブ・フォーマットからイグジットする。「最小化(Minimizing)」アイコン102は、以下でさらに詳しく説明される上述のウェブ・フォーマットの最小化モー

ドをアクティブにする。

【0100】第9C図は、ウェブ・フォーマッタの作業の間に、それにより提供されるメニューを図示する。これらのメニューは、ファイル(File)メニュー103、エディット(Edit)メニュー104、およびウインドウ(Window)メニュー106を含む。ファイルメニュー103の機能は、セーブ(Save)アイコン99、編集(Edit)アイコン97および印刷(Print)アイコン96の機能にそれぞれ一致するところの、セーブ、編集、および印刷のオプションを提供する。イグジット(Exit)オプションもまたファイルメニュー103から抜け出るために提供される。最後にファイル(File)メニュー103は「HTMLファイルを開く(Open HTML File)」オプション107を提供する。このオプションは、ローカルHTMLファイル、即ちネットスケープ(NetScape)からセーブされたファイルなどのユーザのコンピュータに存在するハイパー・メディアファイルを開くか、またはURLをウインドウズのデスクトップヘドラッグ/ドロップすることによって創成されるURLファイルを開く能力を、ユーザに提供する。「HTMLファイルを開く(Open HTML file)」オプション107はまた、他のウェブファイル処理製品により創成されるファイルを、ウェブ・フォーマッタを使用して、RTFファイルとしてフォーマットおよび印刷すること、セーブおよび編集、またはそのどちらかを行うことができるように、それらのファイルを開くために必要な手段も提供する。

【0101】エディット(Edit)メニュー104は「URLのペースト(Paste URL)」オプション109を提供する。この「URLのペースト(Paste URL)」オプション109は、ウェブのページからコピーされたURLアドレス等のペーストバッファの内容を、上記URLフィールド44へペーストするものである。ウインドウメニュー106は、ウェブ・フォーマッタの使用法、維持管理、背景に関する情報をユーザに提供する「ヘルプ(Help Topics)」オプションと、ウェブ・フォーマッタの版本号およびコピーライト表示を容れる・ダイアログ・ボックス(不図示)をユーザに提供する「ウェブフォーマッタについて(About WebFormatter)」オプションとを提供する。ウインドウ(Window)メニュー106はまた「プレファレンス(Preferences)」オプション110も含む。この「プレファレンス(Preferences)」110は、第9D図で示されるプレファレンス・ダイアログ・ボックス112を開く。

【0102】プレファレンス・ダイアログ・ボックス112は、ウェブ・フォーマッタを構成化しあるいは再構成化するために使用される。第9D図で示されるが、プレファレンス・ダイアログ・ボックス112は、「ミニマイズ・ビュー(Minimize View)」オプション113、「一般(General)」オプション114、および「使用WWWブラウザ(WWW Browser to use)」オプション115を含

む。「ミニマイズ・ビュー(Minimize View)」オプション113は、ウェブ・フォーマッタのグラフィック・ユーザ・インターフェースを最小モードで配列するように設定させることができる。2組のオプションが提供されている。第1の組は「印刷(Print)」、「編集(Edit)」および「セーブ(Save)」とを含む。これらのオプションは、第9B図のプリント(Print)アイコン96、エディット(Edit)アイコン97、およびセーブ(Save)アイコン99に対応する。これらのオプションのひとつに接するリストボックスに、チェックマークが現れると、そのオプションのためのアイコンが、たとえばプリントアイコン、エディットアイコン、および、またはセーブアイコン等の最小化モードで画面に表示される。二つ以上のオプションを同時に選択することができる。これに関して、第9E図は、最小化モードにあるウェブ・フォーマッタのための、グラフィック・ユーザ・インターフェースの典型例116を示す。

【0103】第9D図に戻ると、「ミニマイズ・ビュー(Minimize View)」オプション113は、また、「行(Row)」と「スタック(Stack)」のオプションも含む。これらのオプションは、最小化モードに在るウェブ・フォーマッタにおいて、「行(Row)」を選択して水平に、または「スタック(Stack)」を選択して垂直に表示するように設定するというグラフィック・ユーザ・インターフェースを提供できる。これらのオプションのひとつだけは同時に選択することができる。前記の例として、グラフィック・ユーザ・インターフェース116は、一列のアイコンに対応する。

【0104】「使用WWWブラウザ(WWW Browser to use)」オプション115は、ウェブ・フォーマッタと共に、どのWWWブラウザを使用すべきかを決定する。説明のように、Netscape、Internet ExplorerおよびMosaicはブラウザ・オプションとして好適に提供されているが、しかしながら他のブラウザ・オプションもまた提供できる。要求があれば、これらのオプションのうち1個だけは同時に選択されることもできる。デフォルトのブラウザ・オプションはNetscape Navigatorである。

【0105】「一般(General)」オプション114は、「ブラウザと共にスタート(Auto-startwithbrowser)」オプション117、「最小視でオープン(Open in minimized view)」オプション118、「__ページ以上の印刷の前に警告(warn before printing more than __ page s)」オプション119、および、「__MBs以上の印刷の前に警告(Warn before saving more than MBs)」オプション120とを含む。「ブラウザと共にスタート(Auto-startwithbrowser)」オプション117は、ウェブ・ブラウザが、アクティブのときに、ウェブ・フォーマッタが自動的に呼び出されるようにする。このオプションが選択されない(それがデフォルトである)場合は、ウ

インドウ環境でウェブ・フォーマットのアイコンにダブルクリックすることによって、ウインドウ起動メニューからウェブ・フォーマットを選択することによって、またはURLをウェブ・ブラウザから、上で詳述されるウェブ・フォーマットのアイコンへドラッグおよびドロップすることによって、ウェブ・フォーマットは開かれる。「最小視でオープン(Open in minimized view)」オプション118が選択されると、最小化モードでウェブ・フォーマットを開く。しかしながらデフォルトはフル機能モードである。「__ページ以上の印刷の前に警告(warn before printing more than __ pages)」オプション119および「MBs以上の印刷の前に警告(Warn before saving more than MBs)」オプション120は、フォーマットされたドキュメントのセーブされるページの数、およびそれらのページによりそれぞれ使用されるメモリスペースの量を、ユーザが制御できるようにする。これらの両方のオプションのためのデフォルトは、与えられるべき警告が出ないというものである。一般(General)オプションの2個以上が同時に選択されることができるのは言うまでもない。

【0106】プレファレンス・ダイアログ・ボックス112はまた、ユーザが選択したプレファレンスを取り消すキャンセル(Cancel)ボタン121および、ユーザが選択したプレファレンスを確認するオーケー(OK)ボタン122を含む。上の説明のように、ウェブ・フォーマットは、プレファレンス・ダイアログ・ボックス112を通じて、最小化モードに直接入力するように構成されるか、さもなければユーザが第9B図のミニマイジングアイコン102を通じて最小化モードに入力することができる。また上で言及したが、第9E図は最小化モードにあるウェブ・フォーマットのためのグラフィック・ユーザ・インターフェース116の一例を示す。このグラフィック・ユーザ・インターフェース116は、ユーザがウェブを探索している間に、浮動(floating)インターフェースとして表示される。従ってユーザがウェブのページを見るときに、ユーザはまたグラフィック・ユーザ・インターフェース116を見ることになる。グラフィック・ユーザ・インターフェース116(グラフィック・ユーザ・インターフェース43で示されたアイコンに構造的にも機能的にも一致するアイコンを第9E図で含む)の適切なアイコンをクリックすることによって、ユーザは現在のウェブのページをとらえて、処理し、そのウェブのページをRTFファイルの形でフォーマットし、そしてそのRTFファイルをセーブし、編集およびまたはプリントすることができる。別の方法では、ユーザはウェブ・ブラウザからURLをドラッグし、それをアイコンのひとつにドロップすることができる。

【0107】ユーザは、適当なマウスボタンをダブルクリックすることによって、最小化モードにあるウェブ・フォーマットを再配列することができる。この実行によ

り、プレファレンス・ダイアログ・ボックス112に一致するようなプレファレンス・ダイアログ・ボックスがディスプレイ2に現れる。従って、そのユーザはウェブ・フォーマットの構成を希望通りに変更することができる。もしユーザが、最小化モードからフル機能モードに入力したい場合には、ユーザは、第9E図の最大化(Maximizing)アイコン117をクリックする必要があるだけである。

【0108】第10図はウェブ・フォーマットの作業を説明するフローチャートである。ウェブ・フォーマットはステップS1000で起動される。上記のように、起動はウインドウ環境で、ウェブ・フォーマットにダブルクリックすることによって行うことができる。ウェブ・フォーマットがどのように構成化されたのか、即ちフル機能モードであるのか、さもなければ最小化モードであるのかに応じて、グラフィック・ユーザ・インターフェース43に類似のグラフィック・ユーザ・インターフェースか、またはグラフィック・ユーザ・インターフェース116に類似のグラフィック・ユーザ・インターフェースかのどちらかがステップS1000で表示される。以下では、ウェブ・フォーマットのデフォルトモードはフル機能モードであるために、グラフィック・ユーザ・インターフェース43のそれと同類のグラフィック・ユーザ・インターフェースがステップS1000で表示されることとする。

【0109】次にステップS1001で、ウェブ・フォーマットは、上記プレファレンス・ダイアログ・ボックス112、およびオプション・ダイアログ・ボックス62を通して構成化される。ユーザが、以前に設定されたウェブ・フォーマットについての上記構成を変更したくなければ、この行程は必要ではない。ステップS1002で、ドキュメント・フォーマット・データは、上記フィールド44と46~49に入力される。より具体的に言うと、ユーザは、URL(またはファイル名)をURLフィールド44へ入力する。以下に説明されるが、ウェブ・フォーマットは、この情報を使用してそのURLに格納されたウェブのページを処理し、ウェブ・フォーマットの前記構成ならびにフィールド46~49に入力されたデータに基づいてRTFファイルを創成する。

【0110】ステップS1003で、前述のウェブ・リーダー34に類似のウェブ・リーダーが実行される。このウェブ・リーダーはステップS1004で、ワールド・ワイド・ウェブ等のネットワークに接続する。次にステップS1005で、URLまたはファイル名が入力されたかどうかが決定される。上述のように、ウェブ・フォーマットの好適な実施形態では、URLだけを入力することもできる。しかしながら、ウェブ・フォーマットの別の実施形態はファイル名のエンタリーを可能にするので、URLアドレスにあるファイルとは別のファイルを処理する方法についての説明がなされることになる。

【0111】URLがフィールド44に既に入力されている場合は、処理はステップS1006へ進む。ステップS1006ではウェブ・リーダは、URLアドレスにより指定されるハイパー・メディア・ドキュメント（たとえばホームページ）にアクセスする。ステップS1007では、ウェブ・フォーマッタはウェブ・リーダに、ハイパー・メディア・ドキュメントを横断するよう命令する。従ってウェブ・フォーマッタはウェブからURLアドレスを選択し、そのアドレスをコンテナ76に格納する。必要なアドレスのすべてが一旦は選択され、プリント等の処理機能が起動されると、ウェブ・フォーマッタは、コンテナ76のアドレスに格納されるデータを、メモリ5にダウンロードする。ウェブ・フォーマッタは、それから、オプション・ダイアログ・ボックス62で設定される構成情報に基づいて、そのダウンロードされたデータから、予め決定されたデータを抽出し、その抽出されたデータをメモリ5に格納する。このようにしてたとえば、オプションウィンドウ62にある「テキストのみ(textonly)」オプション72が行われているなら、ダウンロードされたデータからテキストだけが抽出される。処理はそれからステップS1011へ進む。

【0112】これとは異なって、ステップS1005で、HTMLソースファイルのためのファイル名が入力されるなら、ウェブ・フォーマッタはウェブ・リーダに、ファイルの第1のサイトにアクセスするように命令する。ステップS1008およびS1009で、そのサイトが横断されて、データが抽出され、上述のステップS1007と同じ方法で格納される。それからステップS1010で、ウェブ・フォーマッタは、さらに多くのサイトがHTMLソースファイルにリストされているかどうかを決定する。そのファイルにさらに多くのサイトがリストされている場合には、フローはステップS1008に戻り、次のサイトがアクセスされる。サイトがそれ以上ない場合には、処理はステップS1011に進む。

【0113】ステップS1011で、ウェブ・フォーマッタは、前に設定されたフォーマット情報に従って、その抽出されたデータを処理する。たとえばコラム・フィールド48が多重(multiple)に設定されている場合は、その抽出されたデータは、多重コラムを有するドキュメントの形にフォーマットされることになる。この処理は、プリント(Print)アイコン96、エディット(Edit)アイコン97、またはセーブ(Save)アイコン99のうちの1つを起動することにより始められ、そして、この処理は、第1の実施形態における処理と同様のものであって、たとえば、そのドキュメントを平坦化し、フォーマット情報に基づいてそのドキュメントをフォーマットするものである。従って、簡潔にするためその詳細な説明は省略される。

【0114】コンテナ内にURLが格納されているドキュメントが一度ダウンロードされ、予めセットされたフォ

ーマットおよび構成に従ってフォーマットされ、ステップS1011でRTFファイルに変換されると、そのRTFファイルはステップS1012で出力される。別の方式では、グラフィック・ユーザ・インターフェースのどのアイコンが起動されていたかに依って、そのRTFファイルは編集されるか、さもなければセーブされるようにしてもよい。

【0115】本発明は図示による特定の実施形態に関して説明されたものである。本発明が、上記実施形態およびその修正に限定されるものではないこと、並びにこの明細書の精神および範囲の内にある限り、種々の変更および修正が可能であることが理解されよう。

付録1

ユーザプロファイルのサンプル

このユーザ・プロファイルはwindows.iniのファイル形式で作成されている。

[Defaults]

Count=4

Title=MyDailyPaper

[1]

Heading=NewsInBrief

Site=1

Section=FrontPage

MaxLevels=5

Maxpages=10

MaxKBytes=2000

Date=today

Print=level0

Template=1

[2]

Heading=SportsInBrief

Site=2

Section=Sports

MaxLevels=0

MaxPages=10

MaxKbytes=200

KeywordFilter="Football"AND"49ers"

Date=today

Print=level0

Template=1

[3]

Heading=MoneyMatters

Site=1

Section=Business

MaxLevels=1

MaxPages=100

MaxKBytes=20000

KeywordFilter="Computer"OR"hardware"OR"Software"

Date=today

Print=all

```

Template=2
[4]
Heading=SriLanka
Site=3
Section=HotNews
MaxLevels=1
MaxPages=100
MaxKBytes=20000
Date=today
Print=1caves
Template=2
サイト・プロファイルのサンプル
#Legend:
#%W-dayoftheweek
#%s-sectionpartofURL
[Defaults]
Count=3
[1]
Title=SanJoseMercuryNews
Username=mwickram
Password=cannon
StartData=StartHeadlines
EndData=EndHeadlines
HomePage=http://www.sjmercury.com/
SectionURL=http://www.sjmercury.com/%S.htm
SectionCount=9
Section1=FrontPage
Section2=International
Section3=National
Section4=Local&State
Section5=EditorialsCommentary
Section6=Business
Section7=Sports
Section8=Living
Section9=Entertainment
[1.Sections]
FrontPage=front
International=intl
National=natl
Local&State=loc
Editorials&Commentary=edit
Business=biz
Sports=spts
Living=liv
entertainment=ent
[2]
Title=TheSanFranciscoChronicle
HomePage=http://www.sfgate.com/chronicle/
SectionURL="http:www.sfgate.com/cig-bin/chronicle/
article-list.cgi?%/S:/chronicle/today"

```

```

SectionCount=5
Section1=News
Section2=Business
Section3=Sports
Section4=Editorial
Section5=Datebook
[2.Sections]
News=News:MN
Business=Business:BU
10 Sports=sports:SP
Editorial=Editorial:ED
Datebook=Datebook:DD
[3]
Title=TheDayNews
Homepage=
http://www.landa.net/lakehouse/anc1Web/dailynew/
SectionURL="http://www.lanka.net/lakehouse/anc1Web/
/dailynew/%W/WS.html"
SectionCount=12
20 Section1=Business
Section2=Editorial
Section3=Features
Section4=Foreign
Section5=Letters
Section6=InBrief
Section7=HotNews
Section8=Probes
Section9=Military
Section10=Politics
30 Section11=Obituaries
Section12=Sports
[3.Sections]
Business=business/intro
Editorial=editorial/final
Features=features/intro
Foreign=foreign/intro
Letters=letters/final
InBrief=inbrief/intro
HotNews=hotnews/intro
40 Probes=proves/intro
Military=military/intro
Politics=politics/intro
Obituaries=obiturai/intro
Sports=sports/intro
付録 2
検索、抽出、印刷のための基準についての文法
サーチのための最大レベル: MaxLevels=<#>
-1: 全てのレベルを検索する
0-n: n レベルまで検索する
50 ドキュメントの最大ページ: MaxPages=<#>

```

n: n ページを超えない最後のドキュメント
ドキュメントの最大サイズ: MaxKBytes=<#>
n: n キロバイトを超えない最後のドキュメント
除外ルール:
Date=today|lessthan<#>
today: 今日掲示された記事のみを検索する
lessthan<#>;n: n 日以上経過していない記事のみを検索する
Retrieve=all|nosubdir|nothisdir|thissiteonly
all: 他のサイトからのページを取り出すことを許す
nosubdir: サブディレクトリへのURLを除外
nothisdir: このディレクトリへのURLを除外
thissiteonly: このサイトからだけのページを取ってくる
キーワードサーチ:
KeywordFilter=<keyword>(AND|OR|NOT)<keyword>:
そのキーワードの組み合わせを含むページのみを累積する
KeywordRank=<#>;n:
ファジー論理を用いてKeywordFilter内のキーワードの
組み合わせに従ってページをランク付けし、上位 n 個の
ページを保持する
KeywordAuthor=<author>:
このauthorにより著作されたページのみを累積する
ExcludeType=ads|nonEnglish
ads: 広告を除外する
nonEnglish: 英語でない記事を除外する
平坦化ルール: Print=all|leaves|level=<#>
all: その線形ドキュメント内のツリーにある全てのノードを含める
leaves: その線形ドキュメント内のツリーにある全ての葉を含める
level=<#>;n: その線形ドキュメント内のツリー n 番目のレベルまでを含める
フォーマット化ルール: Template=<#>
n: デフォルトもしくはユーザテンプレート番号 n に従って印刷
付録 3
記述モジュール
付録 3 A
〈個人用ニュース・プロファイル・エディタ・モジュール〉このプロファイル・エディタはユーザ・プロファイルへのアクセスを統御し、CProfileMgrクラスによって表される。また、ユーザ・プロファイルのローディング及びセーブをも統御する。プロファイル・エディタにより提供されるサービスは:
BOOLNewProfile(CStringfileName);
- そのfilenameにより与えられる新規プロファイルを創成する
BOOLOpenProfile();

- デフォルトのプロファイルをオープンする
BOOLOpenProfile(CStringfileName);
- 当該名前前のプロファイルをオープンする
CProfileEntry*GetFirstEntry();
- 次のプロファイルのエントリをロードして返す
CProfileEntry*GetNextEntry();
- 次のプロファイルのエントリをロードして返す
BOOLWriteEntry(CProfileEntry&entry);
- 当該プロファイル内の新規エントリをセーブする
各々のプロファイルエントリはCProfileEntryクラスにより表された抽出仕様と出力仕様を含む。その方法は:
CURLGetSiteId();
- そのプロファイル・エントリに含まれるサイトIDを返す
CExtractionSpecGetExtractionSpec();
- そのプロファイル・エントリに含まれる抽出仕様を返す。抽出仕様は、サーチ用のキーワード、レベルに対する限界、ページ、サイズ (キロバイト) を含む
COutputSpecGetOutputSpec();
- そのプロファイル・エントリに含まれる出力仕様を返す。出力仕様はフォーマット化命令、ツリー横断ルールを含む
〈ウェブ・リーダー・モジュール〉CWebPageのクラスは、インターネットブラウザへのインタフェースを定義し、実際のウェブのページを表す。また、ウェブページを取って来、そのウェブページ内の他のURLへのリンクや参照を抽出し、ウェブページの内容を維持することになる役目を有する。その方法は:
30 BOOLLoad();
- URL、ユーザ名、パスワードを用いてウェブページを取ってくる
BOOLParse();
- ウェブページ内のデータを解析して、リンクのリストを作成し、相対URLから絶対のURLに変換する
CURLList*GetLinks();
- 当該ウェブページ内のリンク・リストを返す
CPageData*GetData();
- 当該ウェブページに含まれる実際のテキストデータを返す
40 voidFilterContent();
- サイトのデータに従ってタイトル及び他の情報を抽出する
CStringGetTitle();
- サイトのデータに従ってタイトル及び他の情報を返す
CStringGetAuthor();
- 当該ウェブページの作者を返す
intGetSize();
- そのデータのサイズをキロバイトで返す
50 CNetworkクラスはOLE機能を内包させ、インターネット

45

ブラウザとの通信を提供する。

CStringGetUsername();

- 現在セットされているユーザ名を決定する

voidSetUsername(LPCTSTR);

- そのCNetworkオブジェクト内に現在のユーザ名をセットする

CStringGetPassword();

- 現在セットされているパスワードを決定する

voidSetPassword(LPCTSTR);

- そのCNetworkオブジェクト内に現在のパスワードをセ 10 ャットする

voidClose();

- 全てのアクティブな接続を切り、そのCNetworkオブジェクトをリセットする

shortRead(BSTR* pBuffer, short iAmount);

- 当該ブラウザにより検索されたデータを読み出す

longGetStatus();

- 現在のロードの状態を照会する

BOOLOpen(LPCTSTR pURL, short iMethod, LPCTSTR pPostData, long iPostDataSize, LPCTSTR pPostHeaders);

- ネットワークからURLの取り出しを開始する

CStringGetErrorMessage();

- 発呼者に内部生成のエラーメッセージを提供する

shortGetServerStatus();

- サーバにより報知されたエラー状態を決定する

longGetContentType();

- 現在のロードの内容の長さ（全バイト量）を返す

CStringGetContentEncoding();

- 現在のロードのMIME符号化方法を返す

CStringGetExpires();

- このロードにより検索されたデータが最早有効でない時を返す

CStringResolve(LPCTSTR pBase, LPCTSTR pRelative);

- 絶対値URL（十分に調べられた）を生成する

BOOLIsFinished();

- ロードが完了したか否かを決定する

shortBytesReady();

- 発呼者に読み出されるべきバイト数を通知する

〈サイト・ドライバ・モジュール〉サイト・ドライバ・モジュールは、ウェブ・リーダーにサイト情報を提供することになる。サイト・ドライバは機能的にはプロファイル・エディタに類似しており、CSiteDriverクラスにより表される。提供されるサービスは::

BOOLNewProfile(CString fileName);

- 当該fileNameを与えられた新規プロファイルを創成する

BOOLOpenProfile();

- デフォルト・プロファイルをオープンする

BOOLOpenProfile(CString fileName);

- 当該名前のプロファイルをオープンする

46

CProfile*GetFirstSite();

- 当該最初のサイトのエントリをロードして返す

CProfile*GetNextSite();

- 次のサイトのエントリをロードして返す

BOOLWriteEntry(CProfile&entry);

- 当該プロファイルの新規エントリをセーブする

intNumberOfSites();

- 当該プロファイルに指定されたサイトの数を返す

サイト・プロファイル内のエントリは、そのサイトの基礎URLに関する情報、ニュースソースのタイトル、そのサイトへのアクセス方法に関する情報、セクションデータなどの種々の情報を含み、CSiteEntryクラスにより表される。提供される方法は:

CStringGetURL();

- 当該サイトの基礎URLを返す

CStringGetUsername();

- 当該サイトを求めているユーザ名を返す

CStringGetPassword();

- 当該サイトへのパスワードを返す

20 CStringGetTitle();

- 当該サイトへのタイトルを返す

CStringGetTitle();

- ニュースソースのタイトルを返す

intSectionCount();

- セクションのカウント数を返す

付録 3 B

〈ツリー・マネージャ・モジュール〉ツリー・マネージャは、本プログラムにおける中心的なデータ構造（ウェブページノードのツリーである）を保守するもので、CPageTreeによって表される。CPageTreeはWWWを横断して、前記抽出仕様に従って必要なウェブページを検索し、上記ツリーを構築する。提供される方法は:

CPageTreeNode*GetRoot();

- 当該ツリーのルートノードを返す

BOOLBuild(CURLURL, CExtractionSpec&spec);

- 個人ニュース・プロファイル抽出仕様に従って本ツリーを構築する

ページツリー内の個々のノードはCPageTreeNodeによって表される。提供される方法は:

40 BOOLAddChild(CWebPage*page);

- 子供ノードとウェブページのデータとを加える

CWebPage*GetPage();

- そのノードに含まれるウェブページを返す

intNumberOfChildren();

- そのノードに属する子供の数を返す

BOOLIsLeaf();

- もし葉ノードなら、即ち、子供無しならば、真を返す

ウェブのページツリーを横断するには、CTreeIteratorクラスが異なる横断方法で定義される。提供される方法は:

50

```
voidReset();
- 現在の転送をキャンセルし、状態データをリセットする
CPageTreeNode*GetNextNode();
- 深さ方向における最初のサーチでのツリー内の次のノードを返す
CPageTreeNode*GetNextSibling();
- 幅方向における最初のサーチでのツリー内の次のノードを返す
CPageTreeNode*GetNextLeaf();
- 深さ方向における最初のサーチでのツリー内の次の葉を返す
〈フォーマッタ・モジュール〉 このモジュールへの入力
は、前記ツリーマネージャにより創成されたウェブページ・ツリー、並びに、ユーザプロファイルに含まれる出力仕様である。このフォーマッタは、出力仕様のルールに従ってこのツリーを横断し、最終ドキュメントは、出力仕様内のフォーマット化命令や、ウェブページに含まれるヘッディング、パラグラフやリストなどのフォーマットを用いて定式化される。出力ドキュメントはリッチ・テキスト・フォーマット(RTF)であり、多くのアプリケーションによりアクセス可能である。RTFはテキストに対して進んだフォーマット言語であり、ドキュメント、セクションやパラグラフのフォーマットや、スタイルシート、ヘッダやフッタに、ユニコードに対するサポートを提供する。GIFフォーマットであるウェブ画像についてはサポートはない。サードパーティのライブラリが、GIFからDIBフォーマットへの変換を行うのに買うか、開発することが必要となろう。本プロトタイプは出力としてHTMLファイルを生成する。本フォーマッタはCFormatterクラスで表され、提供される方法は：
BOOLOpenHTMLFile(CStringfileName);
- 出力用に名付けられたHTMLファイルをオープンする
voidCloseHTMLFile();
- そのHTMLファイルをクローズしてセーブする
BOOLPrintHTML(CPageTree&root,COutputSpec&format);
- 当該ルートと出力仕様が与えられると、そのツリーを横断し、当該ウェブページの内容をHTMLフォーマットで印刷する
BOOLOpenRTFFile(CStringfileName);
- 出力用に当該名前前のRTFファイルをオープンする
voidCloseRTFFile();
- 当該RTFファイルをクローズしセーブする
BOOLPrintRTF(CPageTree&root,COutputSpec&format);
- 当該ルートと出力仕様が与えられると、そのツリーを横断し、当該ウェブページの内容をRTFフォーマットで印刷する
BOOLPrint(CPageTree&root,COutputSpec&format);
- 当該ルートと出力仕様が与えられると、そのツリーを
```

横断し、当該ウェブページの内容をデフォルト・プリンタに印刷する

【図面の簡単な説明】

【図1】 本発明に従った個人用ニュース検索システムの外観の斜視図である。

【図2】 図1で説明される個人用ニュース検索システムのブロック図である。

【図3A】 本発明に従って情報の変換が行われる際のウェブの構成を示す図。

10 【図3B】 図3Aに示されたウェブから抽出された抽出データ・ツリーの構成を示す図。

【図3C】 図3Bのツリーから平坦化处理されたドキュメントの構成を示す図。

【図3D】 図3Cの平坦化ドキュメントからフォーマットされたドキュメントを示す図。

【図4】 ウェブを通じてニュース記事を検索するための個人用ニュース・プロファイルが、本発明に従って作られ、あるいは編集される方式を具体的に表すブロック線図で。

20 【図5A】 個人用ニュース・プロファイルがどのように作られ、あるいは編集されるかを説明するフローチャート。

【図5B】 個人用ニュース・プロファイルがどのように作られ、あるいは編集されるかを説明するフローチャート。

【図6】 本発明に従って、ニュース記事がウェブから検索され、個人用ニュース・プロファイルと関連しながらフォーマットされる方式の具象的なブロック線図である。

30 【図7】 ニュース記事がどのようにして、個人用ニュース・プロファイルと関連して（プロファイルに照合して）ウェブから検索されるのかを説明するフローチャート。

【図8】 検索されたニュース記事がどのようにして、個人用ニュース・プロファイルと関連してフォーマットされ、プリント装置インターフェースに送られるのかを示すフローチャートである。

【図9A】 本発明の第2の実施形態と共に使用されるグラフィック・ユーザ・インターフェースを描く図。

40 【図9B】 本発明の第2の実施形態と共に使用されるグラフィック・ユーザ・インターフェースを描く図。

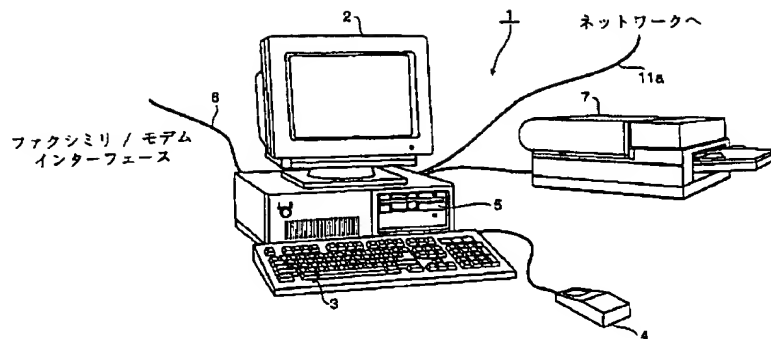
【図9C】 本発明の第2の実施形態と共に使用されるグラフィック・ユーザ・インターフェースを描く図。

【図9D】 本発明の第2の実施形態と共に使用されるグラフィック・ユーザ・インターフェースを描く図。

【図9E】 本発明の第2の実施形態と共に使用されるグラフィック・ユーザ・インターフェースを描く図。

【図10】 本発明の第2の実施形態を説明するフローチャートである。

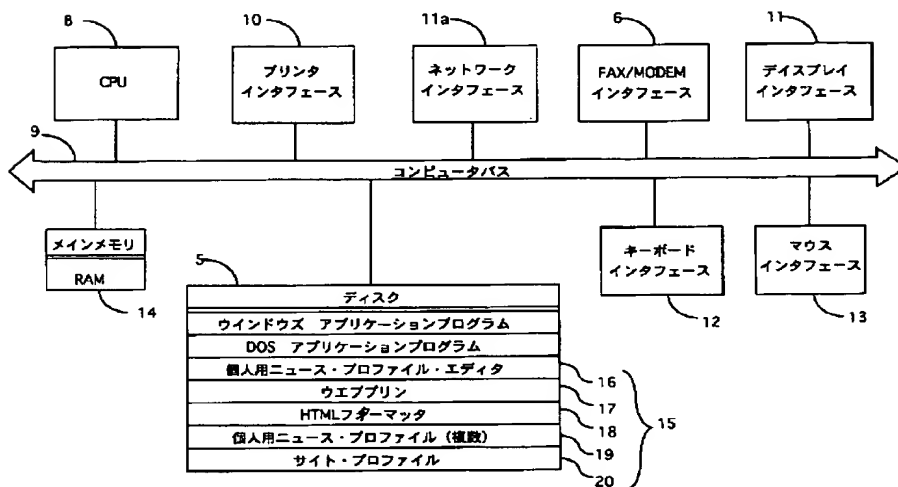
【図1】



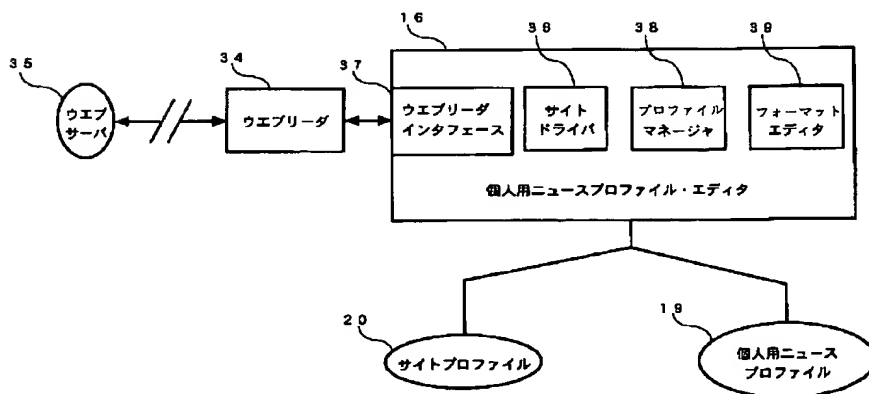
【図3C】

My Paper
サイト#1の名称
インデックス#1
記事Aのタイトル
記事Aのテキスト
記事Bのタイトル
記事Bのテキスト
記事Cのタイトル
記事Cのテキスト
インデックス#3
記事Fのタイトル
記事Fのテキスト
記事Hのタイトル
記事Hのテキスト
サイト#2の名称
:
:
:

【図2】



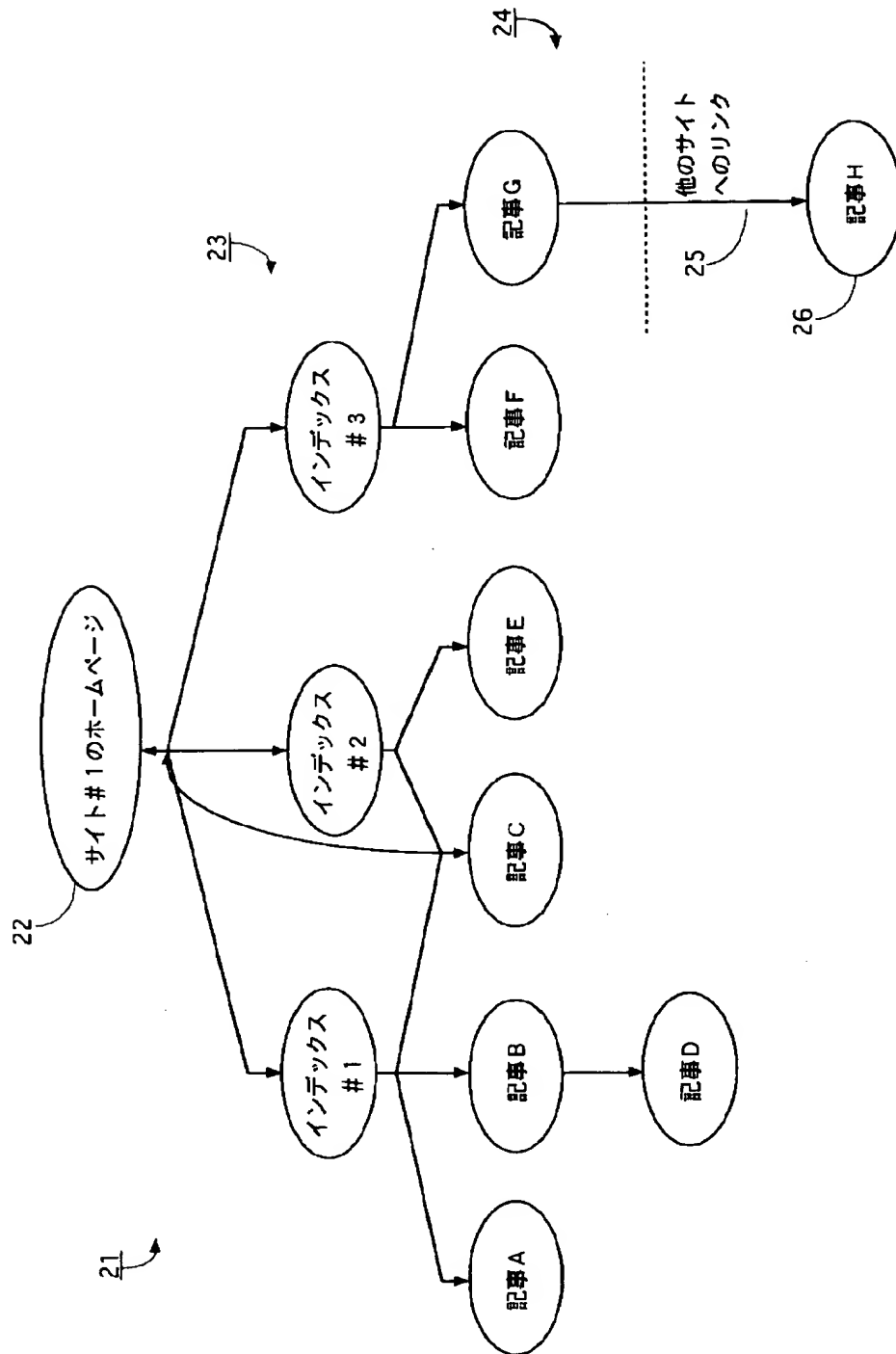
【図4】



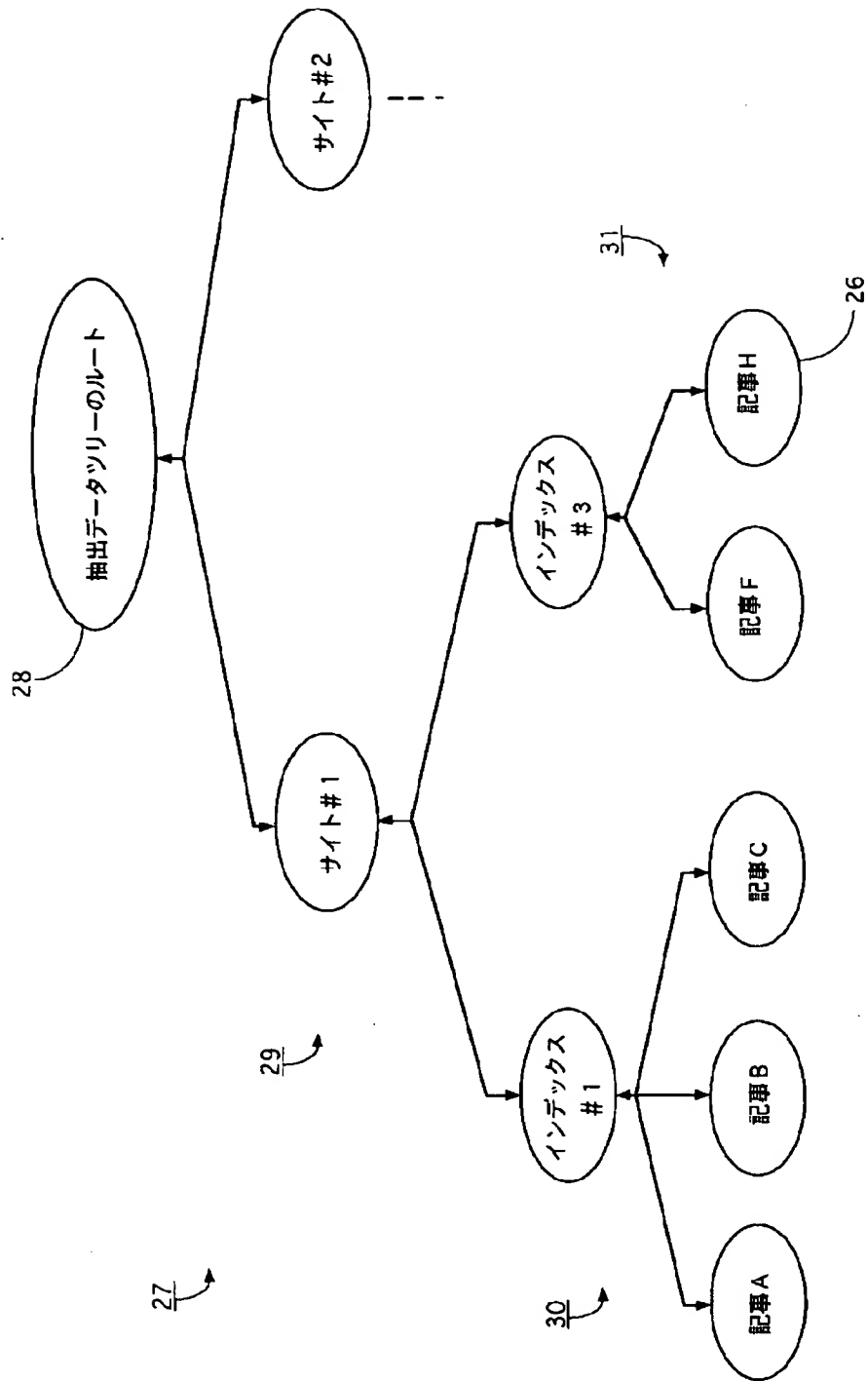
【図9E】



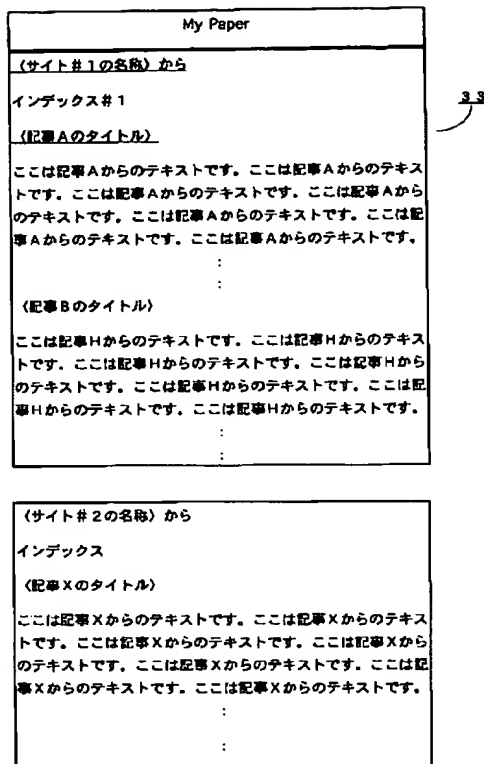
【図3A】



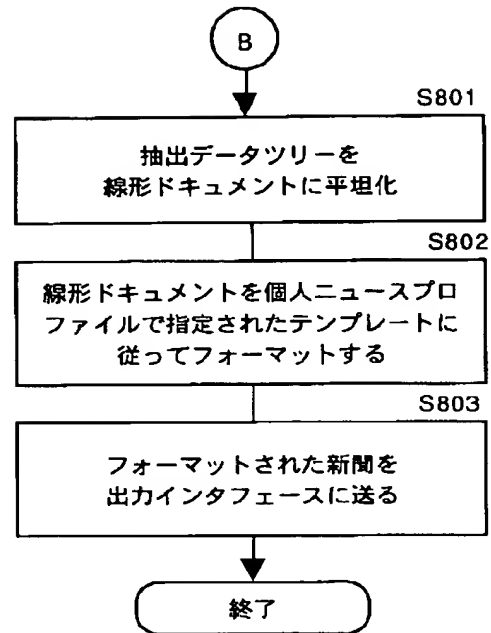
【図3B】



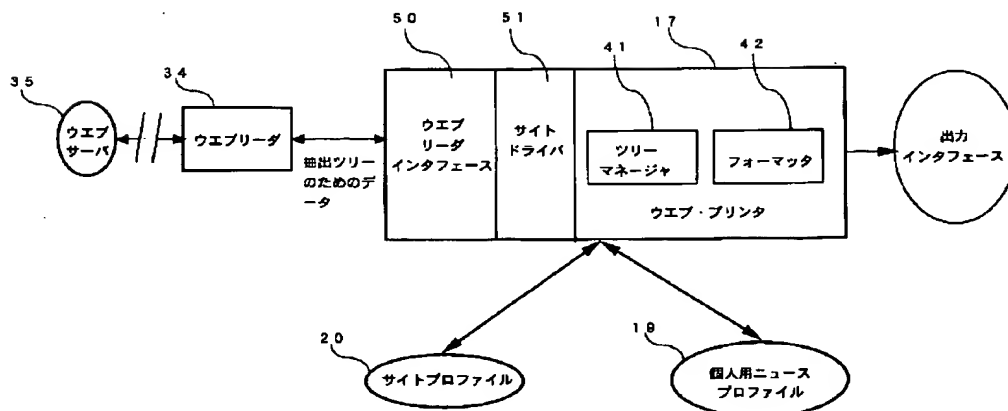
【図3D】



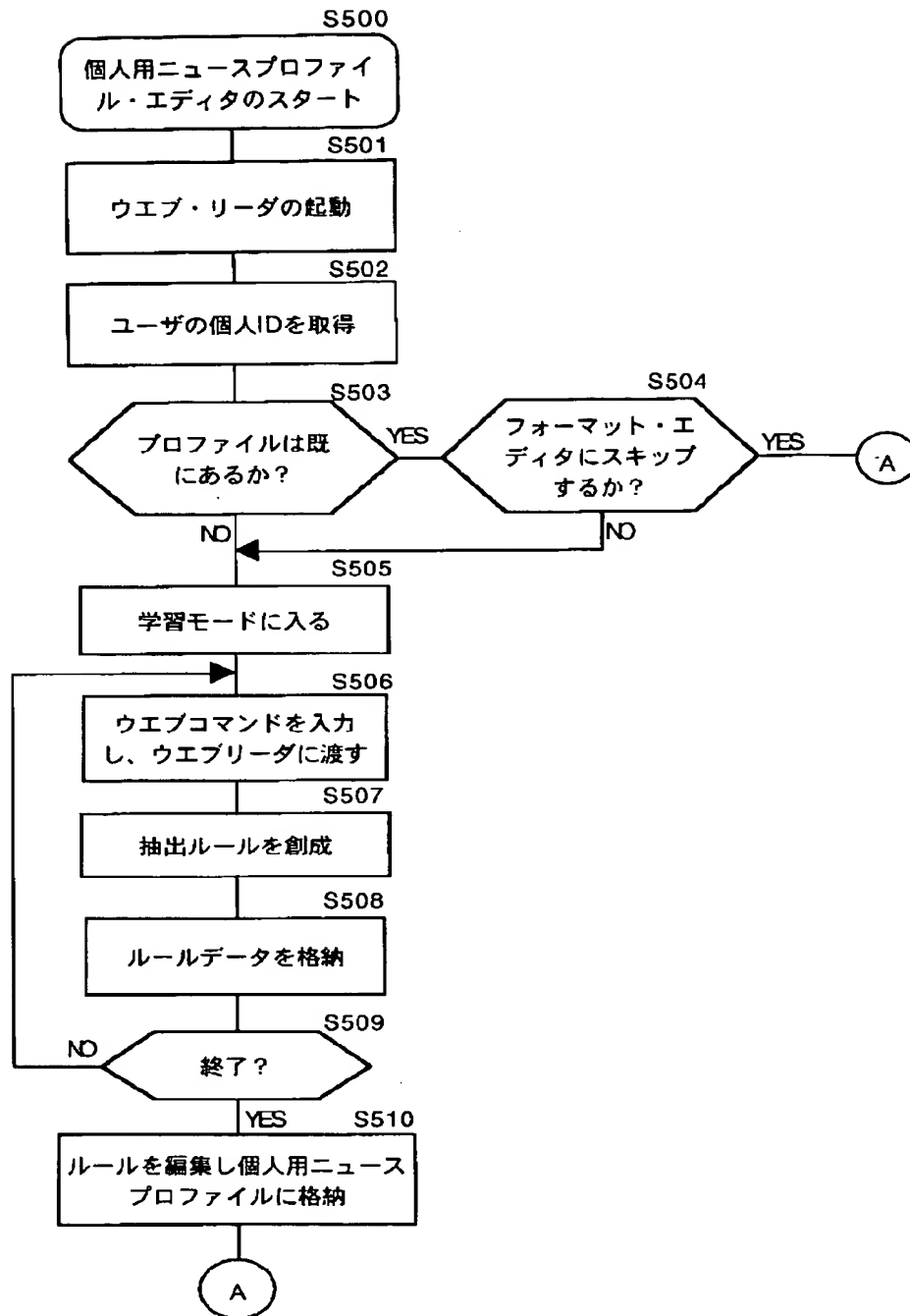
【図8】



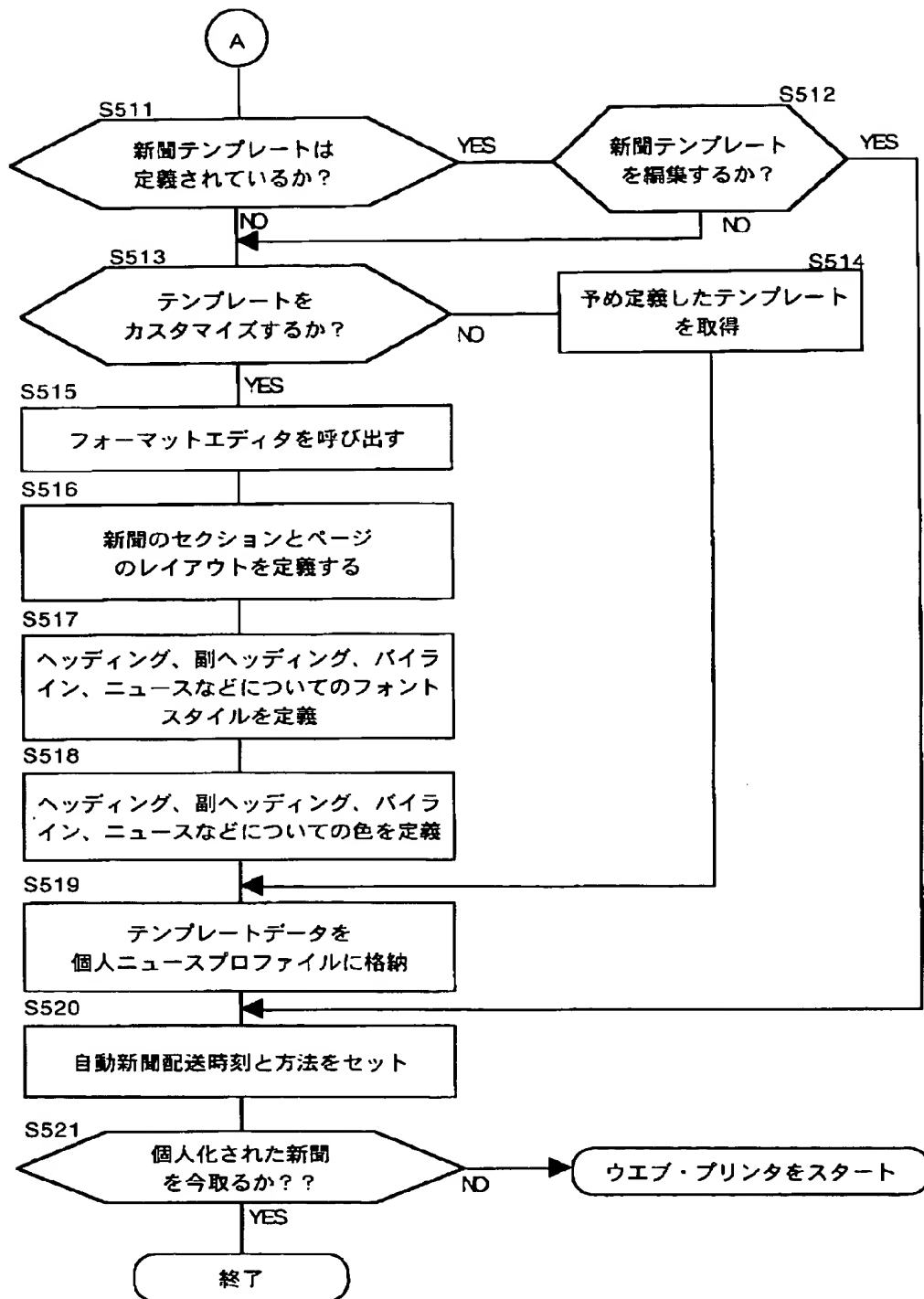
【図6】



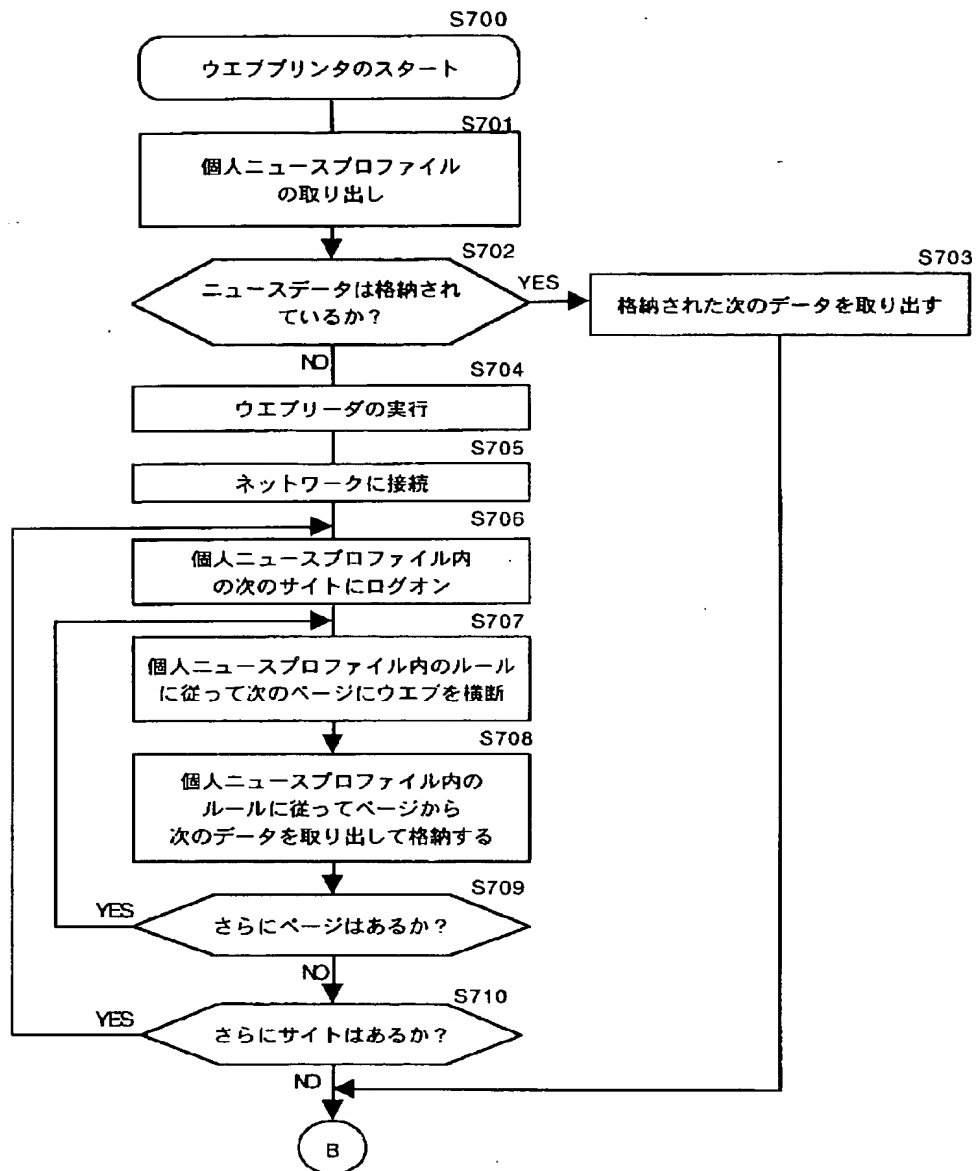
【図5A】



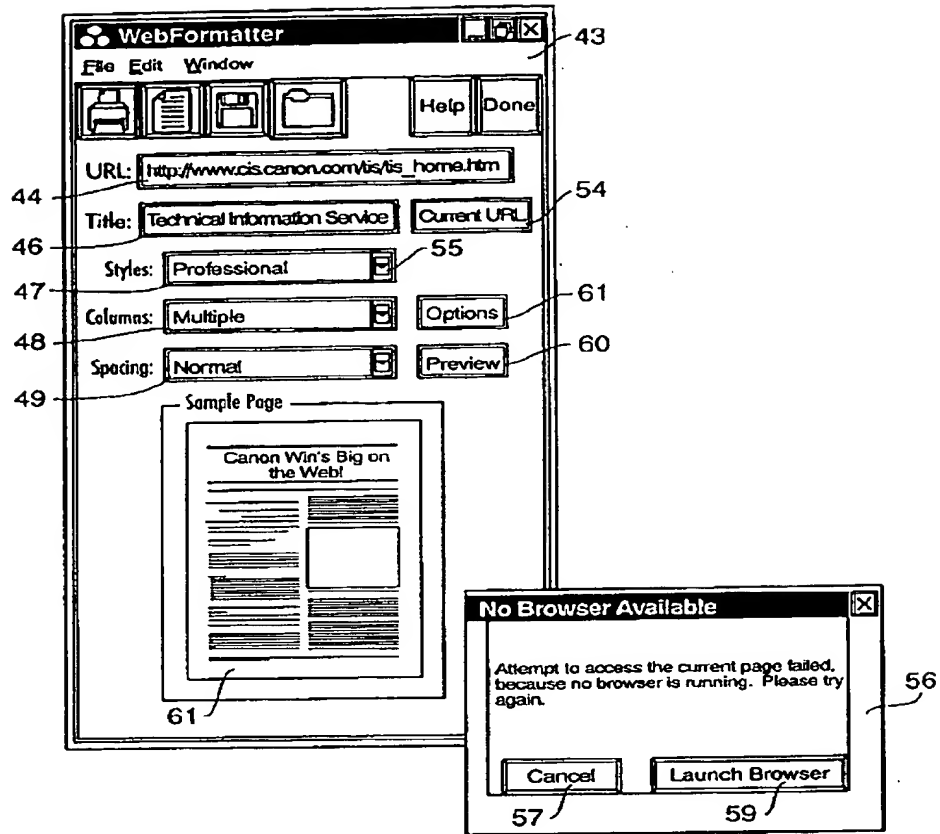
【図5B】



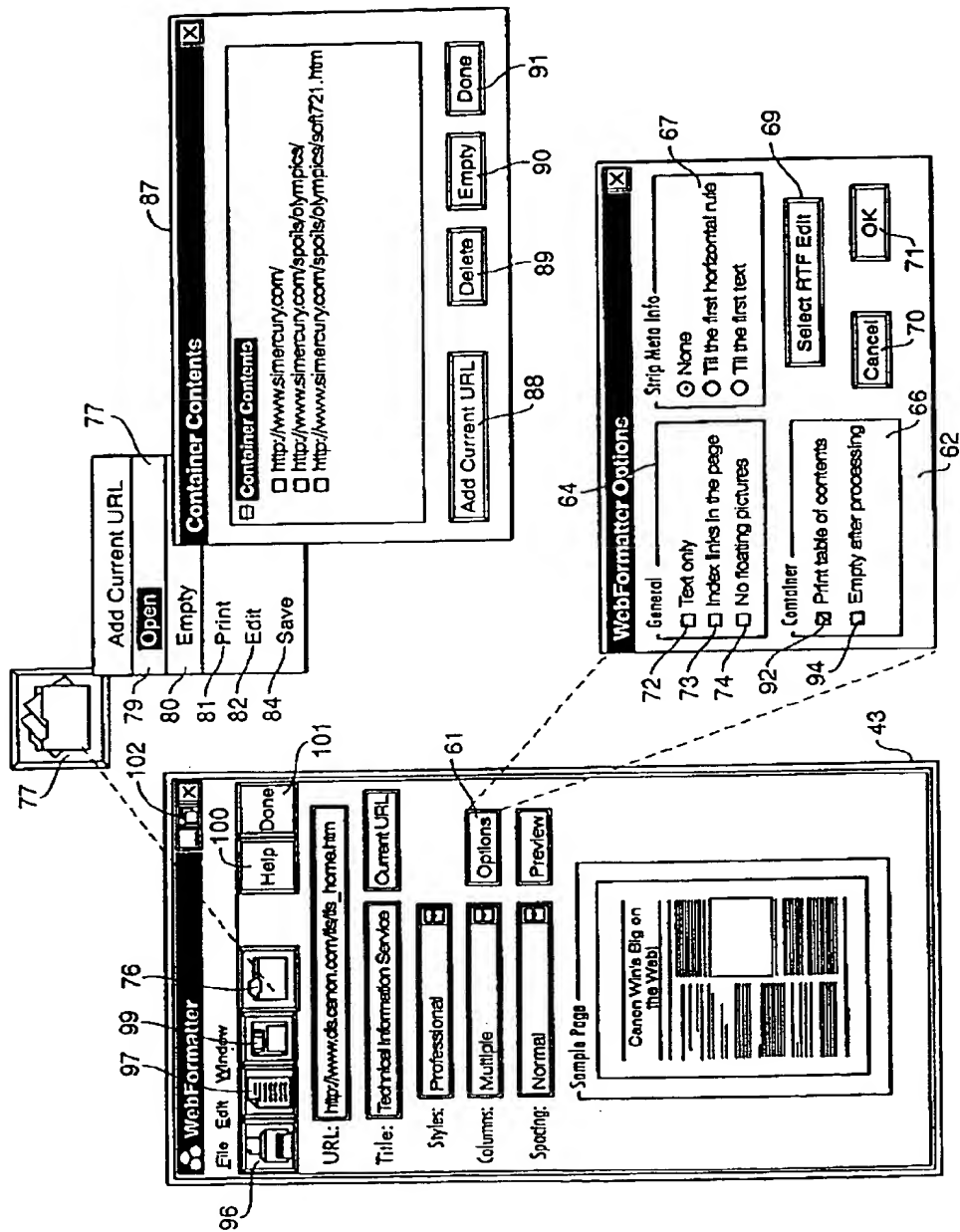
【図7】



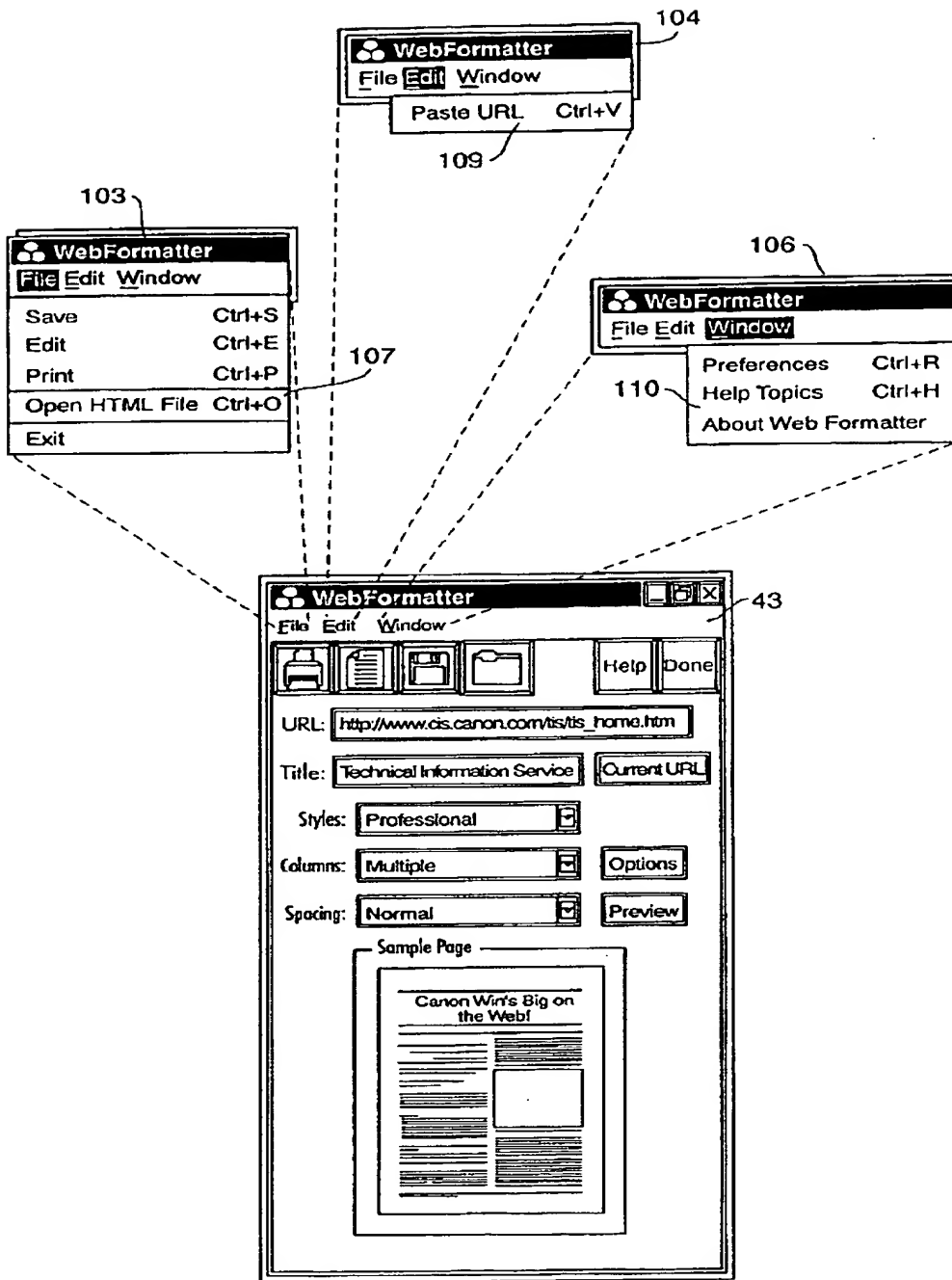
【図 9 A】



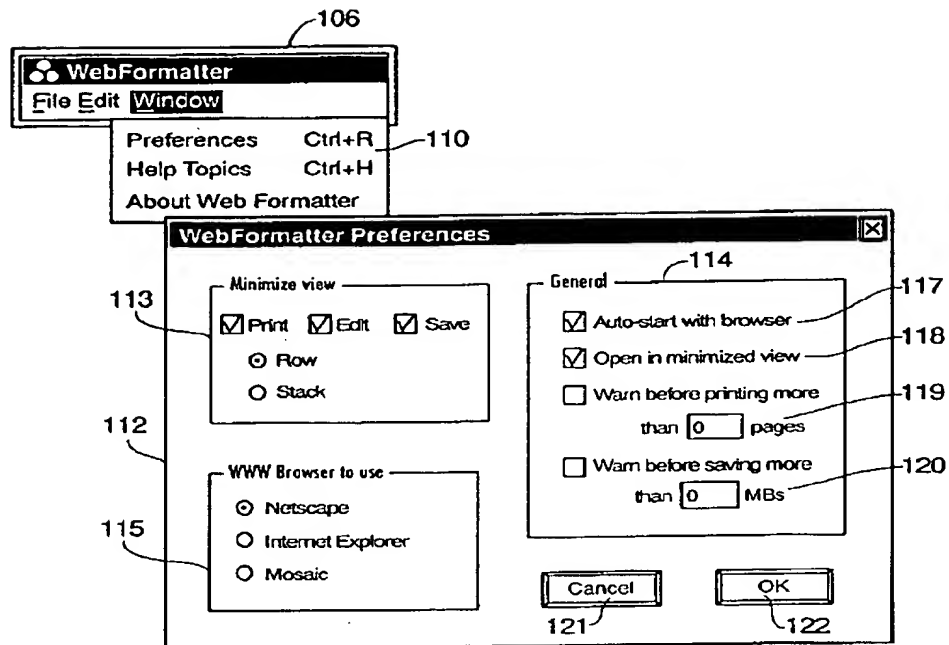
【図9B】



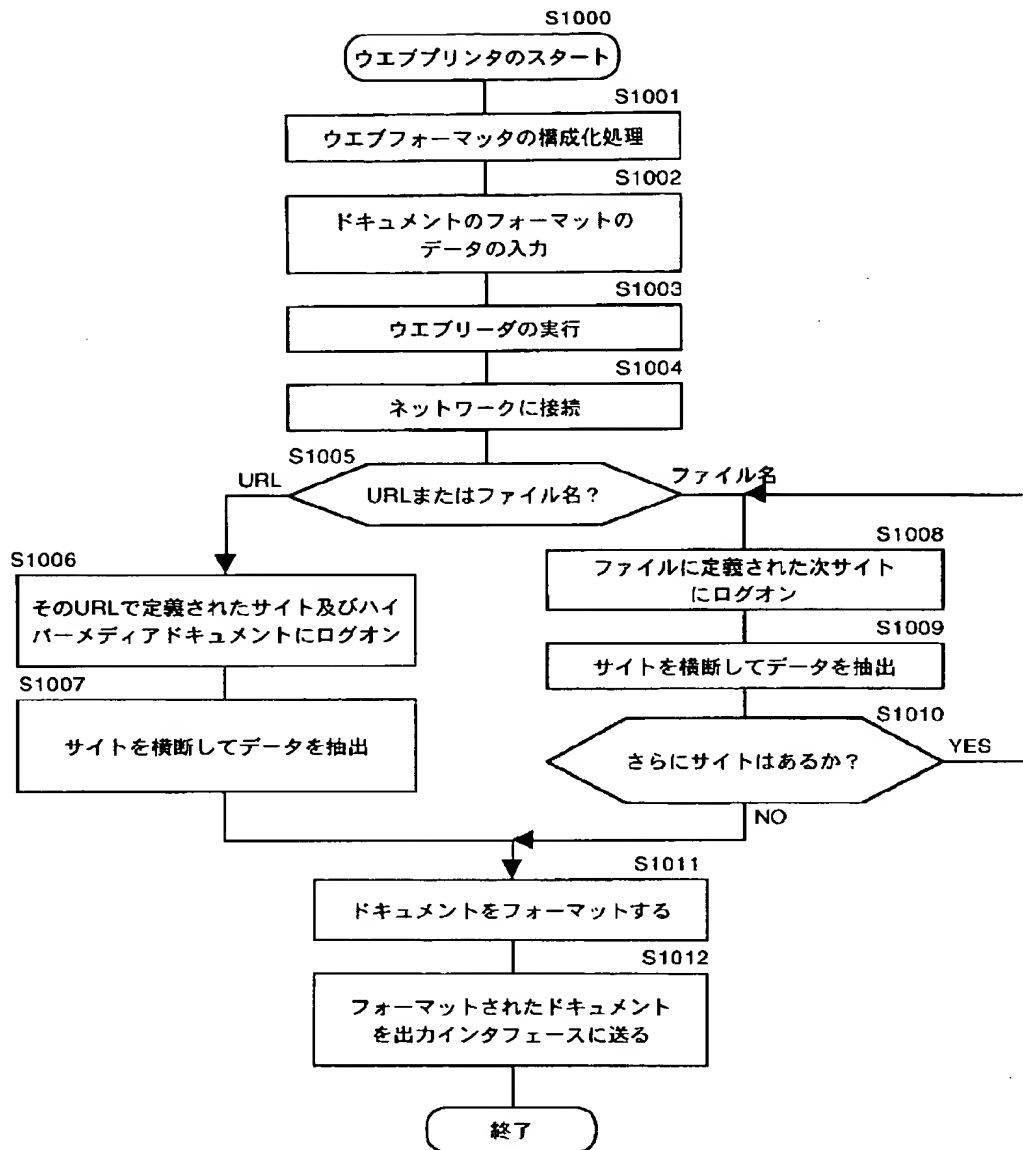
【図9C】



【図 9 D】



【図 10】



フロントページの続き

(72)発明者 マンジュラ ジー・ ウィックラマラネ
アメリカ合衆国 カリフォルニア州
94536, フリーモント, サニースロー
プ ドライブ 333

(72)発明者 パウル エル・ クラーク
アメリカ合衆国 カリフォルニア州
94040, マウンテン ビュー, デル
メディオコート #208, 2700

【外国語明細書】

1. TITLE OF INVENTION
WORLD WIDE WEB NEWS RETRIEVAL SYSTEM
2. CLAIMS
 1. A method for formatting data from at least one hypermedia document, comprising the steps of:
 - an accessing step to access the at least one hypermedia document;
 - a retrieving step to retrieve data from the hypermedia document into an extracted data tree, wherein the data is retrieved based on a structure of the hypermedia document;
 - a flattening step to flatten the extracted data tree into a linear document; and
 - a formatting step to format the linear document into a formatted document.
 2. The method of Claim 1, further comprising the step of printing the formatted document.
 3. The method of Claim 1, wherein said hypermedia document is located on the World Wide Web.
 4. The method of Claim 1, wherein said hypermedia document is located on the Internet.
 5. The method of Claim 1, wherein said hypermedia document is located on an intranet.
 6. The method of Claim 1, wherein said accessing step, said retrieving step, said flattening step, and said formatting step are performed in accordance with a personal-news-profile.
 7. A method of creating a personal-news-profile for retrieving data from a hypermedia-linked computer network, comprising the steps of:

整理番号 = 3 6 P 1 1 1 J

(2)

accessing the hypermedia-linked computer network;
entering a learning mode;
traversing sites on the hypermedia-linked computer network
with commands;
extracting at least one rule from the commands; and
compiling the at least one rule into the personal-news-profile
file.

8. The method of Claim 7, wherein the at least one rule specifies structural characteristics of sites for traversing the hypermedia-linked computer network.

9. The method of Claim 8, wherein the at least one rule also specifies content-based criteria for traversing the hypermedia-linked computer network.

10. A personalization system for creating a personalization profile for a Web site retrieval data retrieval system, the personalization system comprising:

an input device for inputting data and commands to access the World Wide Web;

a connection to the World Wide Web;

a memory for storing a Web reader, the Web reader for accessing the World Wide Web via the connection to the World Wide Web according to commands from the personalization system; and

a processor for launching the personalization system in response to a user command, wherein the personalization system, upon being launched, (1) launches the Web reader, (2) accesses the World Wide Web via the Web reader, (3) enters a learning mode, (4) sends commands to the Web reader to traverse the World Wide Web according to user commands, (5) extracts at least one rule from the user commands, (6) compiles the at least one rule into a personalization profile, and (7) stores the per

整理番号=36P111J

(3)

sonalization profile.

11. A method for retrieving articles from a hypermedia-linked computer network and for formatting the articles into a personalized newspaper, the method comprising the steps of:

retrieving a stored personal-news-profile which comprises address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands;

contacting the site based on address data stored in the personal-news-profile;

downloading articles from the site based on command data stored in the personal-news-profile;

flattening the articles into a linear document; and

formatting the linear document into the personalized newspaper according to layout commands stored in the personal-news-profile.

12. The method of Claim 11, further comprising the step of printing the personalized newspaper.

13. The method of Claim 11, wherein said hypermedia-linked computer network is the World Wide Web.

14. The method of Claim 11, wherein said hypermedia-linked computer network is on the Internet.

15. The method of Claim 11, wherein said hypermedia-linked computer network is on an intranet.

16. The method of Claim 11, wherein the command data for accessing data includes data for selecting articles based on a structure of the site.

17. The method of Claim 16, wherein the command data for accessing data also includes data for selecting articles based on a content of the articles.

18. A World Wide Web site data retrieval system for acces

整理番号=36P111J

(4)

sing at least one Web site, for retrieving data from the Web site, and for formatting the data into a personalized document, the system comprising:

an input device for inputting data and commands to access the World Wide Web;

a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information, wherein the memory also includes process steps to connect to a Web site and to issue commands within the connected Web site;

a connection to the World Wide Web; and

a processor for launching the Web site data retrieval driver in response to a user inputting a command to access the World Wide Web, wherein the Web site retrieval driver, upon being launched, (1) launches the Web reader to connect to the World Wide Web via said connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, wherein the data is downloaded with reference to a linked list so as to avoid hypermedia-links that form loops and so as to avoid repetitious downloading of data that has already been downloaded, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into the personalized document based on the format information.

19. The Web site data retrieval system of Claim 18, wherein the Web site address information, the Web site commands, and the format information stored in the memory form a personalized-news-profile.

整理番号=36P111J

(5)

20. The Web site data retrieval system of Claim 18, further comprising a printer for printing the personalized document.

21. The Web site data retrieval system of Claim 18, where in the personalized document represents a personalized newspaper.

22. The Web site data retrieval system of Claim 18, where in the personalized document represents a personalized magazine.

23. The Web site data retrieval system of Claim 18, where in the personalized document represents a personalized book.

24. Computer executable process steps stored on a computer-readable medium, said steps for accessing World Wide Web sites for retrieving data at the sites and for formatting the data into a personalized document, said steps comprising:

a connecting step to connect to the World Wide Web;

a retrieving step to retrieve user-defined Web site address information, user-defined Web site commands, and user-defined formatting commands;

an activating step to activate a Web reader so as to access a Web site based on the user-defined Web site address information and retrieving data from within the Web site based on the user-defined Web site commands;

a downloading step to download the retrieved Web site data from the accessed Web site into an extracted data tree;

a flattening step to flatten the extracted data tree into a linear document;

a step to repeat the downloading step and the flattening step until all addresses in the user-defined Web site address information have been accessed; and

a formatting step to format the stored data into the personalized document based on the user-defined formatting commands.

整理番号 = 36P111J

(6)

25. The computer executable process steps of Claim 24, further comprising a spooling step to spool the personalized document to a n output device.

26. The computer executable process steps of Claim 25, wherein the output device is a printer.

27. The computer executable process steps of Claim 25, wherein the output device is a display.

28. The computer executable process steps of Claim 24, wherein the user-defined Web site commands include commands for selecting data based on a structure of the Web site.

29. The computer executable process steps of Claim 28, wherein the user-defined Web site commands also include commands for selecting data based on a content of the Web site.

30. An apparatus for retrieving news articles from on-line news services on the World Wide Web and formatting the news articles into a personalized newspaper, the apparatus comprising:

first storage means for storing (1) a personal-news-profile which comprises addresses data and command data for accessing data from a Web site, and (2) newspaper format commands;

retrieval means for retrieving the stored personal-news-profile and accessing data stored therein;

activating means for activating a Web reader to contact a Web site based on address data stored in the personal-news-profile;

downloading means for downloading news articles from the contacted Web site based on command data stored in the personal-news-profile;

second storage means for storing the downloaded news articles; and

formatting means for formatting the stored news articles i

整理番号=36P111J

(7)

into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile.

31. The apparatus of Claim 30, further comprising spooling means for spooling the personalized newspaper to a printer.

32. A method for formatting data from a hypermedia document into a personalized document, comprising the steps of:

a location specifying step to specify a location of the hypermedia document;

a type specifying step to specify the type of the hypermedia document;

a scope specifying step to specify the scope of data to retrieve from the hypermedia document, wherein the scope is based on a structure of the hypermedia document;

a format specifying step to specify a format for formatting the data retrieved from the hypermedia document into the personalized document;

an accessing step to access the hypermedia document found at the location specified in the location specifying step;

a retrieving step to retrieve data from the hypermedia document accessed in the accessing step, wherein the data is retrieved in accordance with the type specified in the type specifying step and in accordance with the scope specified in the scope specifying step; and

a formatting step to format the data retrieved in the retrieving step into the personalized document, wherein the data is formatted in accordance with the format specified in the format specifying step.

33. The method of Claim 32, further comprising a printing step to print the personalized document.

34. The method of Claim 32, wherein the location specified in the location specifying step is a filename.

整理番号=36P111J

(8)

35. The method of Claim 32, wherein the location specified in the location specifying step is a uniform resource locator for the World Wide Web.

36. A method of processing a hypermedia document, comprising the steps of:

accessing the hypermedia document;

extracting addresses from the hypermedia document;

storing the addresses extracted from the hypermedia document in a container in a memory;

activating a processing function to process data stored at the addresses stored in the container;

downloading the data stored at the addresses in the container into the memory;

extracting predetermined data from downloaded data in accordance with predetermined configuration information;

formatting the predetermined data in accordance with predefined formatting settings to generate a formatted document; and

processing the formatted document in accordance with the processing function.

37. A method according to Claim 36, further comprising a step of previewing the formatted document prior to processing the formatted document.

38. A method according to Claim 37, further comprising the steps of:

changing the formatting settings after previewing the document and before processing the formatted document in accordance with the processing function;

re-activating the processing function; and

re-formatting the data in accordance with changed formatting

整理番号=36P111J

(9)

ng settings to generate the formatted document.

39. A method according to Claim 36, wherein the addresses are stored in the container in the order that the addresses are input into the container; and

wherein the processing function processes the predetermined data in the order that the addresses are stored in the container.

40. A method according to Claim 39, further comprising the step of rearranging the addresses stored in the container by dragging and dropping the addresses within the container.

41. A method according to Claim 36, further comprising a step of inputting the formatting settings and configuration information via a graphical user interface.

42. A method according to Claim 41, wherein the graphical user interface comprises plural processing icons, one of which activates the processing function.

43. A method according to Claim 42, wherein the graphical user interface is displayed in plural modes.

44. A method according to Claim 43, wherein the plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons.

45. A method according to Claim 44, wherein the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document.

46. An apparatus for processing a hypermedia document, comprising:

a Web read which accesses the hypermedia document;
means for extracting addresses from the hypermedia document

整理番号 = 3 6 P 1 1 1 J

(1 0)

t;

a memory including a container which stores the addresses extracted from the hypermedia document;

a graphical user interface having processing icons which activate at least one processing function to process data stored at the addresses stored in the container; and

processing means which (1) downloads the data stored at the addresses stored in the container into the memory, (2) extracts predetermined data from downloaded data in accordance with predefined configuration settings, (3) formats the predetermined data in accordance with predefined formatting settings to generate a formatted document, and (4) processes the formatted document in accordance with the processing function.

47. An apparatus according to Claim 46, further comprising previewing means for previewing the formatted document prior to processing the formatted document.

48. An apparatus according to Claim 46, wherein the addresses are stored in the container in the order that the addresses are input into the container; and

wherein the processing function processes the predetermined data in the order that the addresses are stored in the container.

49. An apparatus according to Claim 48, further comprising dragging and dropping means for dragging and dropping the addresses listed in the container in order to rearrange the addresses in the container.

50. An apparatus according to Claim 46, further comprising inputting means for inputting the formatting settings and configuration information via a graphical user interface.

51. An apparatus according to Claim 50, wherein the graph

整理番号 = 3 6 P 1 1 1 J

(1 1)

ical user interface comprises plural processing icons, one of which activates to the processing function.

52. An apparatus according to Claim 51, further comprising display means for displaying the graphical user interface in plural modes.

53. An apparatus according to Claim 52, wherein the plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons.

54. An apparatus according to Claim 53, wherein the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document.

3. DETAILED DESCRIPTION OF INVENTION

Field Of The Invention

The invention relates to a data retrieval system which automatically traverses hypermedia documents on a computer network and automatically retrieves information from those documents based on a match between the structure of the documents and a personalized data retrieval structure. More particularly, the invention can retrieve articles from a news service, from a magazine service, or from a combination of both services which are located on the World Wide Web, a private computer network that supports hypermedia links, or any other hypermedia-linked computer system.

For example, there exists a Web site for retrieving news articles from the New York Times and a Web site for retrieving articles from People magazine. The retrieval system of the invention can traverse through such Web sites and select articles based on a personalized data retrieval structure. The personalized data retrieval structure may inc

整理番号 = 36P111J(12)

lude commands to retrieve a full text of the front page only, headlines of the business section, headlines of the stock section and sports section, etc. In addition, the personalized data retrieval structure may include content-based rules to retrieve articles with certain keywords, to exclude articles with certain keywords, or to include articles based on a rule-based content analysis. The invention also provides a method for synthesizing all retrieved news articles and printing the synthesized news articles into a newspaper-type format in which each of the articles is arranged based on a user's predefined layout.

While the above example is in the context of the Web, hypermedia documents can reside on other types of networks besides the Web, such as an intranet. An intranet is a private computer network that is not connected to outside computer networks. For example, a company's own computer network could be an intranet with hypermedia documents on it.

For brevity, the following discussion is made with respect to the World Wide Web. However, it should be understood that the invention applies equally well to any type of computer network that contains hypermedia documents, such as an intranet, different hypermedia-linked computer networks that reside on the Internet other than the Web, etc.

A hypermedia document on the Web can span multiple Web sites. Such documents can be newspapers, news articles, magazines, catalogs, manuals, memoranda, and the like. For brevity, the following discussion is made with respect to sources of news information. However, it should be understood that the invention applies equally well to any other type of hypermedia document.

Description Of The Related Art

The World Wide Web is an on-line source of hypermedia documents containing hypermedia text and images that act as links to other documents, Web sites, etc. As a result, documents on the Web are not org

整理番号=36P111J

(13)

anized sequentially. Rather, a user is automatically linked to other documents or Web sites to complete the viewing of a document by selecting a hypermedia link, such as a text link or an image link, within the document. Accordingly, an entire document cannot be viewed by scrolling through text.

One popular use of the Web is on-line publication and distribution of magazines and newspapers. Currently, many Web news services, such as the New York Times, allow the user to define keywords of interest and to receive news information, daily or hourly, that contains text matching the keywords. The news information can then be delivered to the user's computer via modem or E-mail. However, most Web news site newspapers, like the New York Times, include too much information, most of which has no interest to the user since the information is retrieved based only on a keyword match.

Other sources of news information are provided through information suppliers like "Individual Inc." Individual Inc. supplies users with a brief summary of the top twenty most relevant articles based on a user's predefined keywords. This subscription news service allows the user to specify five to ten areas of interest based on keywords, which are then prioritized by the user. The information service searches the Web for magazines and newspapers which contain any of the keywords. Based on the keyword searches, twenty of the most relevant articles are selected, compiled into a brief one-page summary, and transmitted to the user via facsimile for the user's review. However, in order to review an entire document rather than the summary, the user must log onto a specific Web site containing the document in order to retrieve and review the document.

There are yet other services which permit the user to personalize a newspaper to be displayed at the user's terminal by storing li

整理番号 = 3 6 P 1 1 1 J(1 4)

links to various news articles from various news sources on the Web. For example, CRAYON "Create Your Own Newspaper" permits a user to select specific sections from among links to over twenty-five different on-line newspapers, and to compose the selections into a personalized newspaper. Using CRAYON, it is possible to compose a personalized newspaper containing, for example, links to the international section of the New York Times, the business section of the Wall Street Journal, and the sports section of the Chicago Tribune. The HTML (hypertext markup language) source file for this newspaper is then stored to mass media storage for later use.

While the foregoing news and information services provide convenient ways to keep updated on the news, they do not allow a user to access and view the news in the way that people naturally read a real-world newspaper. Namely, people naturally read a newspaper by scanning the pages of sections that they find interesting and then reading those articles that grab their attention. In other words, people use a structural approach to decide what pages to look at initially (e.g., the first page of the Business and World sections, and the comics page of the Arts section). They then scan the selected pages for articles.

In sum, conventional news and information services do not allow a user to access data from a hypermedia document on the basis of the structure of the document, and then to format that data in a manner that allows the user to scan and read the data in a natural fashion.

SUMMARY OF THE INVENTION

The invention addresses the above deficiencies in the art by accessing at least one hypermedia document, retrieving data from the hypermedia document into an extracted data tree, with the data retrieved based on a structure of the hypermedia document, flattening the extracted data tree into a linear document, and formatting the linear document

整理番号=36P111J

(15)

into a formatted document.

In another aspect, the invention creates a personal-news-profile for retrieving data from a hypermedia-linked computer network. The hypermedia-linked computer network is accessed, a learning mode is started, the hypermedia-linked computer network is traversed with commands, at least one rule is extracted from the commands, and the rule(s) is compiled into the personal-news-profile.

In yet another aspect, the invention creates a personalization profile for a Web site retrieval data retrieval system. Data and commands are input to access the World Wide Web and a connection is made to the World Wide Web. A Web reader is launched, and the Web reader accesses the Web via the connection. In response to user commands, a learning mode is entered into. Commands are sent to traverse the World Wide Web, and at least one rule is extracted from the commands. The rule(s) is compiled into a personalization profile, which is stored.

In yet another aspect, the invention retrieves articles from a hypermedia-linked computer network and formats the articles into a personalized newspaper. A stored personal-news-profile is retrieved. The personal-news-profile includes address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands. The site is accessed based on address data stored in the personal-news-profile, and articles at the site are downloaded based on command data stored in the personal-news-profile. The downloaded articles are flattened into a linear document, and the linear document is formatted into the personalized newspaper according to newspaper layout commands stored in the personal-news-profile.

In yet another aspect, the invention retrieves data from a World Wide Web site and formats the data into a personalized document.

A Web site data retrieval driver which includes a Web reader, stored W

整理番号 = 36P111J(15)

Web site address information, stored Web site commands, and stored format information is accessed. The invention (1) launches the Web reader to connect to the World Wide Web via a connection to the Web, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, wherein the data is downloaded with reference to a linked list so as to avoid hypermedia-links that form loops and so as to avoid repetitious downloading of data that has already been downloaded, (5) stores the Web site data in a linear document, (6) repeats steps 2 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into the personalized document based on the format information.

In yet another aspect, the invention accesses and retrieves data at World Wide Web sites and formats the data into a personalized document. The invention connects to the World Wide Web, retrieves user defined Web site address information, user defined Web site commands, and user defined formatting commands, and activates a Web reader so as to access a Web site based on the user defined Web site address information. The Web reader is used to download data from the Web based on the user defined Web site commands, and the data is downloaded into an extracted data tree. The downloading continues until all addresses in the user defined Web site address information have been accessed. The extracted data tree is flattened into a linear document, and the flattened document is formatted into the personalized document based on the user defined formatting commands.

In yet another aspect, the invention retrieves news articles from on-line news services on the World Wide Web and formats the news articles into a personalized newspaper. The invention stores a persona

整理番号 = 36P111J

(17)

l-news-profile which comprises addresses data and command data for accessing data from a Web site and newspaper format commands, retrieves the stored personal-news-profile and accesses the data stored therein, activates a Web reader to contact a Web site based on address data stored in the personal-news-profile, downloads news articles at the contacted Web site based on command data stored in the personal-news-profile, stores the downloaded news articles, and formats the stored news articles into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile.

In yet another aspect, the invention formats a hypermedia document into a personalized document. A location of the hypermedia document is specified, a type of the hypermedia document is specified, a scope of data to be retrieved from the hypermedia document is specified, wherein the scope is based on a structure of the hypermedia document, and a format is specified for formatting the data retrieved from the hypermedia document into the personalized document. The hypermedia document found at the specified location is accessed, data is retrieved from the hypermedia document in accordance with the specified hypermedia document type and in accordance with the specified scope, and the data is formatted into the personalized document in accordance with the specified format.

In yet another aspect, the invention is a system for processing a hypermedia document. The system accesses the hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from downloaded data in accordance with predetermined configuration information.

整理番号= 3 6 P 1 1 1 J

(18)

on. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

In preferred embodiments, the system inputs the formatting settings and configuration information via a graphical user interface.

The graphical user interface comprises plural processing icons, one of which activates the processing function. By virtue of the graphical user interface, a user can interactively set a document's format and change that format should a change be desired.

In particularly preferred embodiments, the graphical user interface is displayed in plural modes. The plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons. Typically, the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document. By displaying the graphical user interface in plural modes, the present invention facilitates operation of the invention during browsing of the hypermedia document.

This summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiments thereof in connection with the attached drawings.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 is a view showing the outward appearance of a representative embodiment of the invention. Shown in Figure 1 is computing equipment 1, such as a MacIntosh or an IBM PC or a PC-compatible comput

整理番号=36P111J(19)

er, having a windowing environment, such as Microsoft Windows. Provided with computing equipment 1 is display screen 2, such as a color monitor or a monochromatic monitor, keyboard 3 for entering text data and user commands, and a pointing device such as mouse 4 for pointing and for manipulating objects displayed on display 2. Computing equipment 1 also includes a mass storage device such as disk drive 5. Image data can be input into computing equipment 1 from a variety of sources such as a network interface 11a or from external devices via facsimile/modem interface 6. Network interface 11a is used to connect computing equipment 1 to a local area network (LAN) or to a wide area network (WAN) such as the World Wide Web.

Figure 2 is a detailed block diagram showing the internal construction of computing equipment 1. As shown in Figure 2, computing equipment 1 includes central processing unit (CPU) 8 interfaced with computer bus 9. Also interfaced with computer bus 9 is printer interface 10, fax/modem interface 6, display interface 11, network interface 11a, keyboard interface 12, mouse interface 13, main memory 14, and disk drive 5.

Main memory 14 interfaces with computer bus 9 so as to provide random access memory storage for use by CPU 8 when executing an application such as personal news profile editor 16 or Web printer 17. More specifically, CPU 8 loads these software applications from disk drive 5 into main memory 14 and executes the software applications out of main memory 14. In accordance with user instructions, stored application programs are activated which permit processing and manipulation of data. Typically, the software applications stored on disk drive 5, such as personal-news-profile editor 16, Web printer 17, and HTML formatter 18, have been stored on disk drive 5 by downloading the software applications from a computer-readable medium such as a floppy disk or CD ROM, or by do

整理番号=36P1111J (20)

unloading the software applications from a computer bulletin board.

Disk drive 5 stores data files which can include text files and image files, in compressed or uncompressed format, and stores software application files such as those noted above. The software application files include Windows applications, DOS application, and personal news retrieval files 15. Personal news retrieval files 15 include personal-news-profile editor 16, Web printer 17, HTML formatter 18, personal-news-profile(s) 19, and site profile(s) 20. The detailed functions of personal news retrieval files 15 will be discussed below, after a brief overview of the operation of the personal new retrieval system.

Overview of Document Retrieval

Figure 3, comprised of Figures 3A to 3D, illustrates the operation of a representative embodiment of the invention. Figure 3A is a graphical representation of a typical Web site 21 with news information contained therein. Within Web site 21 is homepage 22 with links to indices such as headings 23, which are in turn linked to articles 24. Some of articles 24 are linked to other articles. As article H 26 resides on another Web site, link 25 is a cross-site link. Link 25 illustrates how a single hypermedia document, represented by homepage 22, can traverse multiple Web sites.

In order to retrieve news from Web site 21, the invention first traverses Web site 21 to retrieve data according to user-defined rules. As will be discussed in more detail below, these rules can be based on the structure of Web site 21, or on the structure of Web site 21 and its contents. The data is retrieved into an extracted data tree, which preserves the organization of the data as shown in Figure 3B, but in which some links are excluded.

The organization of extracted data tree 27 has several features. First, extracted data tree 27 has root 28 which can have child n

整理番号 = 3 6 P 1 1 1 J

(21)

odes for one or more sites 29, which in turn can have index nodes 30 which correspond to indices/headings 23, articles nodes 31, and the like. Second, extracted data tree 27 is a true tree, with no loops (i.e., cyclic paths) therein. For example, Figure 3A shows a loop from homepage 22 to index node #1, to article C, and then back to homepage 22. This loop is removed when creating extracted data tree 27.

Second, the organization of extracted data tree 27 depends on how the Web sites are traversed, and not on the Web sites' actual layouts. Thus, article H 26 appears under index node #3 (under site #1), indicating that the news retrieval system accessed article H 26 from site #1 via cross-site link 25.

Finally, as noted earlier, certain articles have been excluded from extracted data tree 27 due to the structure of Web site 21 or possibly a content of indices/headings 23 and articles 24. For example, articles B and G have been excluded from extracted data tree 27.

According to the invention, extracted data tree 27 is flattened into linear document 32, as shown in Figure 3C, possibly with reference to more exclusion rules. Linear document 32 is simply a continuous document with information from extracted data tree 27 embedded therein.

Finally, linear document 32 is formatted according to user specified (or default) formatting instructions into formatted document 33, shown as a stylized personal newspaper in Figure 3D. Formatted document 33 has various fonts and/or colors for site labels, indices/headings, articles, and the like. Furthermore, formatted document 33 is broken down into pages.

Note that in alternate embodiments of the news retrieval system, certain stages of the above transformation from Web site 21 to formatted document 33 can be skipped. For example, data from Web site 21

整理番号 = 3 6 P 1 1 1 J

(22)

can be retrieved directly into flattened document 32, as long as a record of the organization of the data is maintained (possibly in a separate linked list) so as to avoid downloading the same article twice and so as to avoid loops in the organization of Web site 21. Alternatively, extracted data tree 27 can be directly formatted into formatted document 33.

In any case, the basic operation of the invention remains the same: the news retrieval system traverses a hypermedia document on the Web, extracts data according to user-defined information, and formats the data into a personalized newspaper.

As mentioned in the above discussion, various user-defined rules and other information (such as formatting information) are involved in the news retrieval process. That user defined information is stored in personal-news-profile(s) 19, the definition of which is described next.

Defining a Personal-News-Profile

Figures 4 and 5 illustrate the process by which personal-news-profile 19 is defined. To create personal-news-profile 19, personal-news-profile editor 16 communicates with personal-news-profile 19, site profile 20, and Web reader 34.

Personal-news-profile 19 contains information as to what sites to access for creating a personalized newspaper, what sections to retrieve from those sites, rules to be used to determine what data to extract from the sections and the articles therein, rules to determine how to exclude links, and newspaper format information. A sample personal-news-profile is shown in Appendix 1.

Site profile 20 includes general site information that is not specific to a particular user. For example, site profile 20 could contain information such as full site addresses, sections within a site, non-user specific passwords, etc. Sample site profiles are shown in App

整理番号 = 3 6 P 1 1 1 J

(2 3)

endix 1. Because general site information is stored in site profile 20, personal-news-profile 19 can refer to the general site information with reference to site profile 20, saving space in the personal-news-profile. For example, as shown in Appendix 1, personal-news-profile 19 can refer to a site number 1. Site profile 20 indicates that site number 1 is the "San Jose Mercury News," with a homepage at "http://www.sjmercury.com/". This construction also centralizes general site information. Thus, if a site address changes, only site profile 20 needs to be changed to update all personal-news-profiles 19 on the system.

Web reader 34 is an application program or program module that communicates with the Web via Web server 35. In response to commands from personal-news-profile editor 16, Web reader 34 will access the Web, traverse hypermedia documents on the Web, retrieve data from the documents, and return the retrieved data to personal-news-profile editor 16.

As shown in Figure 4, personal-news-profile editor 16 includes four modules: site driver 36, Web reader interface 37, profile manager 38, and format editor 39.

Web reader interface 37 interfaces personal-news-profile editor 16 to Web reader 34. Site driver 36 interacts with Web reader 34 via Web reader interface 37 to provide an abstract interface to each individual Web site. More specifically, site driver 36 instructs Web reader 34 to access various Web sites and to retrieve data from those sites.

Thereafter, site driver 36 receives that data and builds site profile 20 therefrom. The data can also be used to update an existing site profile.

In building site profile 20, site driver 36 translates the structure of each accessed Web site to a uniform structure defined in site profile 20, and stores data retrieved therefrom in site profile 20.

整理番号=36P111J

(24)

By translating different Web sites, some of which may have different structures, into a single uniform structure and storing data therefrom in that structure in site profile 20, the present invention facilitates access to information from different Web sites, and thus reduces overall processing time.

Profile manager 38 maintains document templates that specify how to format a personalized newspaper. Predefined document templates exist. In addition, format editor 39 allows a user to specify personalized templates for formatting a newspaper, either by editing existing templates or by creating new ones. In any case, each document template specifies page layout information, font information, style information, colors, etc. for the titles, indices/headings, subheadings, text and the like for a personalized newspaper.

Sample code for personal-news-profile editor 16, site driver 36, and profile manager 38 is included in Appendix 3A.

Figures 5A and 5B are flow diagrams describing the operation of personal-news-profile editor 16 in more detail. Figure 5A shows the operation of personal-news-profile editor 16 in defining the parts of personal-news-profile 19 relating to accessing Web sites and retrieving data from those sites.

In step S500 of Figure 5A, personal-news-profile editor 16 is launched by a user. In step S501, the editor launches Web reader 34. The user's personal I.D. is then retrieved in step S502. If a personal-news-profile already exists for that I.D., step S503 directs flow to step S504, where the user is given the option of skipping to the format editor. Otherwise, personal-news-profile editor 16 enters a "learning mode" in step S505. Once in the learning mode, personal-news-profile editor 16 proceeds to step S506, where it accepts a Web command (i.e., a command to traverse a hypermedia link) from the user and forwards the Web

整理番号=36P111J

(25)

command to the Web reader by means of site driver 36. Site driver 36 maintains a hierarchical log of Web sites visited by Web reader 34. In step S507, personal-news-profile editor 16 creates an extraction rule from the Web command. This rule will allow the news retrieval system to later duplicate the user's selection criteria in browsing (clicking on hyperlinks within) a Web site.

The rule specifies, at the least, structural criteria for duplicating the traversal of the Web site. For example, if a user accesses all articles under a particular index/heading, the rule will specify that all articles under that index/heading should be retrieved.

In one embodiment of the invention, the rule can also include content-based criteria (i.e., keyword-based criteria) accepted from the user. These content-based rules can, for example: (1) require certain words to be in an article, (2) exclude articles with certain words, (3) require certain boolean combinations of words, (4) rank articles that are selected based on structural criteria, with the ranking based on keywords, and then require the selection of the articles with the highest ranking(s), or (5) exclude certain types of articles such as advertisements.

Examples of the syntax for the structural and content-based exclusion rules are shown in Appendix 2. Several different types of rules are shown. Some simply limit the traversal of a Web site to a certain number of links. Others are date and keyword based exclusion rules.

One particularly flexible rule indicates that articles should be ranked based on a keyword analysis and the top scoring articles should be chosen. Other rules include "flattening" rules. These rules control the flattening of the extracted data tree, as will be explained in more detail below.

At the least, the rule includes structural information abo

整理番号 = 36P111J

(26)

ut the user's selection (i.e., first page, first document, all links, et c.), necessary password information, browser commands, and the like. The rule can also include a pointer or a reference to site profile 20 and the appropriate information therein. General (non-user specific) information is used by site driver 36 to maintain site profile 20. In this manner, address information and passwords common to multiple users can be maintained in site profile 20, as discussed above. For example, site driver 36 will store commands or hyperlinks to other documents in a Web page in the rule, but will not store a Web site's full address in the rule. That address information is stored in site profile 20.

In step S508, rule data defining the rule created from a Web command(s) is stored in an extracted data tree such as extracted data tree 27 in Figure 3B. This data tree is a linked list that reflects the organization of the data retrieved from the Web. In step S509, flow returns to step S506 for the next Web command unless the user is done (i.e., the user signs off the Web site), in which case flow proceeds to step S510.

At this point, the creation of the personal-news-profile has proceeded much like the creation of a macro common to word processing programs, except that site profile 20 has been used to minimize storage requirements and to centralize general site information. In order to minimize storage requirements further and in order to make the news retrieval system more flexible and efficient, the extracted rules are now compiled to remove redundant links, multiple visits to the same site, and the like. This occurs in step S510, and the resulting compiled rules become the first part of personal-news-profile 19.

Alternatively, personal-news-profile editor 16 may be invoked as a graphical user interface which allows a user to edit a previously stored personal-news-profile or to specify document composition preferences.

整理番号= 3 6 P 1 1 1 J

(2 7)

rences, for example, by specifying news sites, headline articles only, keywords, etc. In either case, the result is personal-news-profile 19, which comprises a listing of Web site pointers as well as extracted rules for traversing through a Web site or sites.

For a better understanding of the above, sample personal-news-profiles and sample site profiles are provided in Appendix 1 as noted above.

Next, operation proceeds to give the user an option to modify a custom newspaper template, as shown in Figure 58. In step S511, it is determined if a newspaper template has been defined and stored in personal-news-profile 19. If a newspaper template has been defined, step S512 gives the user the option to edit the template or to proceed to step S520. If the user chooses to edit the template or if no newspaper template has been defined, flow proceeds to step S513.

Step S513 gives the user the option of creating a custom template or using a predefined template. If the user wants to use a predefined template, step S514 gets the specified predefined template, which is added to the personal news profile in step S519. Otherwise, flow proceeds to step S515, where format editor 39 is invoked.

Format editor 39 has a graphical user interface that provides the user with a number of formatting options. In step S516, format editor 39 allows the user to define which newspaper sections are to be printed in the newspaper, which Web site's news article are to be placed in each section, and/or how each page is to be laid out. In this regard, the user can specify which Web site's news articles are to be used as a front page, which Web site's news articles are to be used as a business page, which Web site's news articles are to be used as a sports page, etc. In addition, in step S516, the user can define where each index/headline should be listed, as well as what sub-headings should go on each p

整理番号=36P111J(28)

age.

In step S517, format editor 39 allows the user to define the font styles for indices/headings, sub-headings, bylines and actual text of news articles. In step S518, format editor 39 prompts the user to define index/heading colors, title colors, etc. In this regard, layout editor 39 is capable of determining the types of fonts and colors available to the user based on the system's printer capabilities.

Once all of the information is gathered for the custom template, the format editor adds the information to personal-news-profile 19 in step S519. Alternatively, profile manager 38 may also store the custom format as a template in a common area for use by other users. In this case, only a pointer or reference to the custom template is stored in personal-news-profile 19.

In step S520, personal-news-profile editor 16 prompts the user to set an automatic newspaper delivery time and method (i.e., print or store on disk drive 5 for later printing). These settings are added to personal-news-profile 19. More specifically, in the case that a user's computer is continuously supplied with power, the Web news retrieval system can be launched automatically at a designated time. The system will retrieve articles from the Web sites which are listed in personal-news-profile 19. Upon retrieving the news articles, the articles will be formatted based on the newspaper template in personal-news-profile 19.

The formatted personalized newspaper can then be either printed or stored for later viewing. In the case that a time is not set for newspaper delivery, the user can execute the Web news retrieval system program at any time.

Once personal-news-profile 19 has been created, the Web news retrieval system, upon being launched, can traverse Web news sites and build a personalized newspaper by automatically retrieving various new

整理番号=36P111J

(29)

s articles from the Web news sites and print the news articles based on the newspaper template indicated in personal-news-profile 19. A description of how the Web news retrieval system of the invention performs this function is described next.

Retrieving a Document Using a Personal-News-Profile

Figure 6 is a representational block diagram of the manner by which the invention retrieves articles from the Web according to personal-news-profile 19. (Figure 6 also shows the manner by which the retrieved articles are flattened into a linear document and formatted. These functions are discussed in greater detail in the next section of this application.)

As shown in Figure 6, Web printer 17 is responsible for retrieving news articles. Web printer 17 is an end-user application that communicates with personal-news-profile(s) 19, site profile 20, Web reader 34, and output interface 40 in order to perform this function.

Web printer 17 looks at personal-news-profile 19 to determine which Web sites to access and which data to retrieve from those sites. Web printer 17 also looks at site profile 20 for general site information. According to the information in personal-news-profile 19 and site profile 20, Web printer 17 instructs Web reader 34 to connect to the Web via Web server 35 in order to access various Web sites and to retrieve data from those sites. Web reader 34 sends the retrieved data to Web printer 17, and Web printer 17 uses the data to build an extracted data tree. As will be discussed in greater detail in the next section of the application, Web printer 17 then flattens the extracted data tree into a linear document and formats the linear document for output via output interface 40.

As shown in Figure 6, Web printer 17 includes four program modules: Web reader interface 50, site driver 51, tree manager 41, and

整理番号=36P111J(30)

formatter 42.

Web reader interface 50, like Web reader interface 37 described above, interfaces Web printer 17 to Web reader 34.

Site driver 51 accesses site profile 20 and personal-news-profile 19 and provides data stored therein to Web reader 34. As noted above, Web reader 34 uses that data to access various Web sites and to extract data therefrom. As noted above, this retrieved data is used by Web printer 37 to build an extracted data tree.

Tree manager 41 manages the extracted data tree. In this regard, tree manager 41 keeps track of the organization of the retrieved data in the extracted data tree. This allows Web printer 17 to avoid accessing the same article twice, to avoid unnecessarily re-visiting a Web site, and to avoid getting caught in a cycle (loop) in the organization of a hypermedia document on the Web. Alternatively, tree manager 41 could store the data in blocks (as opposed to directly in a data tree) with reference to a linked list that provides the same functionality as the extracted data tree. Sample code for tree manager 41 is included in Appendix 3B.

Formatter 42 is responsible for flattening the extracted data tree into a linear document and formatting the linear document into a personalized newspaper. Formatter 42 performs these functions in accordance with the print criteria and format information (i.e., newspaper template) indicated in personal-news-profile 19. Sample code for formatter 42 is included in Appendix 3B.

In more detail, Figure 7 is a flow diagram describing how Web printer 17 uses Web reader 34 to traverse the Web according to personal-news-profile 19 and to retrieve articles from the Web according to the profile, excluding unwanted data.

The Web printer starts in step S700. In step S701, Web pr

整理番号=36P111J

(31)

inter 17 retrieves either a user designated personal-news profile or a default personal news profile stored in disk drive 5 using site driver 51. In this regard, because computer equipment 1 may be used by more than one user, there may be one or more personal news-profiles stored on the equipment, one of which will be designated as the default. Upon retrieving the designated personal-news-profile, in step S702 Web printer 17 determines whether any news data has been previously stored to disk drive 5 (for example, by a previous automatic news delivery) or if news articles should be retrieved using personal-news-profile 19.

In the case that news data does exist on disk drive 5, in step S703 the stored news data is retrieved and flow proceeds to step S801 of Figure 8, discussed in more detail in the next section. On the other hand, if no stored news data exists, Web printer 17 invokes Web reader 34 in step S704. Note that this is the same Web reader 34 as discussed above with respect to defining a personal-news-profile.

Upon being invoked, Web reader 34 connects to Web server 35 in step S705, which provides a connection to a network, such as the World Wide Web. Web printer 17 then provides Web reader 34 with an address for the first Web site to be visited based on information retrieved from personal-news-profile 19. Once connected to the desired Web site in step S706, Web printer 17 provides Web reader 34 with commands/links for traversing the Web to the next Web page containing information that personal-news-profile 19 indicates should be retrieved. Web reader 34 traverses the Web according to this information in step S707.

In step S708, Web reader 34 retrieves the desired information and sends it to Web printer 17 according to the rules in personal-news-profile 19. Thus, data exclusion occurs in this step. The rules in personal-news-profile 19 specify structural and content-based criteria for excluding data from the personalized newspaper. The structural rules

整理番号=36P111J(32)

limit the retrieved information on the basis of the structure of the Web site accessed by Web reader 34. The content based rules limit the retrieved information on the basis of its content. As mentioned above with respect to creating a personal-news-profile, examples of the syntax of the retrieval rules in personal-news-profile 19 are included in Appendix 2.

In addition to rule-based exclusion, media-type exclusion occurs in step S708, wherein data of a media type that can not be printed is excluded from the extracted data tree. For example, movie and sound data can be excluded.

Web printer 17 stores the retrieved data in disk drive 5 (or in main memory 14) in the extracted data tree managed by tree manager 41. Alternatively, the data could be stored in blocks with reference to a linked list, as discussed earlier. In step S709, Web printer 17 returns to step S707 to complete retrieving all information from Web pages at the Web site. In step S710, upon completing a traversal of one Web site, Web printer 17 uses tree manager 41 to compare the sites remaining in personal-news-profile 19 with the site organization information in the extracted data tree to determine if more sites need to be visited. In the case that more Web sites need to be visited, step S710 returns flow to step S706 and news articles are retrieved in the same manner as discussed above. On the other hand, if all of the Web sites listed in personal-news-profile 19 have been visited and all of the articles retrieved, flow proceeds to step S801 in Figure 8.

Flattening and Formatting the Retrieved Data

Figure 8 is a flow diagram showing how the extracted data tree is flattened and formatted. The configuration of the invention is the same as when retrieving data from the Web (shown in Figure 6). In fact, the flattening and formatting processes can occur, at least to a li

整理番号=36P111J

(33)

limited extent, concurrently with the data retrieval process.

In step S801 of Figure 8, the extracted data tree is flattened. This simply means that the organization of the data is converted from an extracted data tree to a linear document. This step provides the opportunity for excluding more data from the personalized newspaper, for example by only including nodes of the data tree into the flattened document. This exclusion process is controlled by the flattening rules in personal-news-profile 19.

After the data is flattened into a linear document, the data is formatted in step S802 according to the template indicated in personal-news-profile 19. The definition of this template, which is either a pre-defined template or a custom template, was discussed earlier. Finally, in step S803, the formatted and fully personalized newspaper is sent to output interface 40. This interface could be printer interface 10 to printer 7, display interface 11 to display 2, or even modem/fax interface 6.

Second Embodiment: The HTML Formatter

The second embodiment of the invention is a system for processing a hypermedia document. The system accesses the hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from the downloaded data in accordance with predetermined configuration information. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

The second embodiment of the invention is depicted as HTML

整理番号=36P111J(34)

formatter 18, noted in Figure 2. An example of HTML formatter 18 is WebFormatter, manufactured by Canon Information Systems, Inc. The second embodiment will be described with respect to WebFormatter. It should be noted, however, that HTML formatter 18 is not limited to the WebFormatter embodiment, and that various alternative embodiments within the spirit and scope of the following description are possible.

WebFormatter is stand-alone utility software that can be used in conjunction with different Web browsers, such as Netscape, Mosaic and Internet Explorer. In short, WebFormatter extracts data from a Web page, strips out extraneous data from the extracted data, and reformats the data into a formatted document. The formatted document can then be printed, stored in an RTF (Rich Text Format) file, or edited in any RTF compatible editor, such as MS Word, WordPerfect, Wordpad, etc.

WebFormatter can be activated from a windowing environment, such as Microsoft Windows(r). From such a windowing environment, WebFormatter can be activated by double-clicking on a WebFormatter icon (not shown) in a start-up window, selecting WebFormatter from the Windows start menu, dragging a URL (uniform resource locator) icon (not shown) from a Web browser and dropping it into the WebFormatter icon, or by automatically invoking WebFormatter when the Web browser is started.

Unlike the first embodiment of the invention described above, WebFormatter does not use a predefined personal-news-profile to specify criteria for creating a particular type of document from one or more Web pages. Rather, WebFormatter relies upon user-specified criteria to create a particular type of document, such as a newspaper or the like, from one or more Web pages. These criteria are input interactively by a user via a graphical user interface.

As described in more detail below, WebFormatter operates in two modes - a minimized mode and a fully-functional mode. In the mini

整理番号=36P111J

(35)

mized mode, WebFormatter's graphical user interface is essentially a floating print button, which is displayed concurrently with displayed Web pages. By virtue of this feature, as a user explores the Web, the user can process, format, and print out Web pages by merely clicking on the floating print button.

In its fully-functional mode, WebFormatter's graphical user interface provides spaces for a user to enter a URL address of a Web page to be processed, enter a personal title for the document, select a format for the document, preview a formatted first page of the document, and either print the document, save the document as an RTF file, or view/edit the document using an RTF editor. The graphical user interface for the fully-functional mode will be described first, since it is from that interface that the user can enter the minimized mode.

Figure 9A shows graphical user interface 43 for WebFormatter's fully-functional mode. Graphical user interface 43 is displayed on display 2 upon first activation of WebFormatter. As with any interactive windowing software application, a user interacts with graphical user interface 43 by means of mouse 4 (by pointing and clicking) and keyboard 3.

As shown in Figure 9A, graphical user interface 43 includes fields 44 and 46 to 49, through which a user can specify the URL address of a document to be formatted and the format of that document. Beginning with URL field 44, a user enters the URL address (e.g., http://www.cis.canon.com/tis/tis_home.htm) of a Web page to be processed by WebFormatter. There are several different ways for the user to enter the URL address. The user can (1) type the address directly into URL field 44, (2) copy the URL address in the Web browser and paste the URL address into URL field 44, (3) drag the URL address from the Web browser onto graphical user interface 43 or onto the WebFormatter icon, or (4) click on C

整理番号=36P111J

(36)

current URL button 54.

With regard to Current URL button 54, if a user clicks on Current URL button 54, WebFormatter locates the active Web browser and queries the Web browser for the address of the current Web page. Thereafter, the Web browser provides the address of the current Web page to WebFormatter, which places the address in URL address field 44. If URL button 54 is activated and no Web browser is currently running, WebFormatter displays dialog box 56, shown in Figure 9A.

As shown, dialog box 56 includes Cancel button 57 and Launch Browser button 59. Cancel button 57 cancels a user's request to input a URL address into URL address field 44 via Current URL button 54. Launch Browser button 59, on the other hand, launches a Web browser specified in WebFormatter. As noted below, WebFormatter is configured beforehand with predefined information including a Web browser to be used with WebFormatter. Configuration of WebFormatter will be described in more detail below.

In alternative embodiments of WebFormatter, a filename can also be entered into URL address field 44. For example, in these alternative embodiments, if a user wishes to format a hyper linked manual into a book-like format, the user enters the filename into URL address field 44. Thereafter, WebFormatter proceeds through the file in the same manner as through specified Web pages in order to reformat the hyper-linked manual as desired.

Returning to graphical user interface 43, title field 46 enables a user to enter a personalized title for a formatted document. The title may be typed directly or pasted into title field 46.

Formatting fields 47 to 49 define the format of a document to be output by WebFormatter. Options for the different formatting fields can be accessed by clicking on a scroll bar, such as scroll bar 55.

整理番号=36P111J

(37)

of a respective formatting field. Each of these fields is described in detail below.

Styles field 47 provides four options for formatting an output document. These styles relate to characteristics of an output document such as size of headers, margins, etc. The style options include Contemporary, Formal, Fun and Professional. The invention, of course, is not limited to these four style options, and other styles can be added as desired.

Columns field 48 defines the number of columns in a formatted output document. Two columns options are available - Single and Multiple; however, the invention is not limited to these two options. The Single option, as might be expected, formats the document into a single column. The Multiple option, on the other hand, formats the document into a predetermined number of columns. In preferred embodiments of the invention, the multiple option is set to two columns; however, any number can be set.

Spacing field 49 defines the spacing between lines in a formatted output document. Three options are provided in WebFormatter, but other options can be added as desired. These three options are Condensed, Normal and Easy To Read, with Condensed being the least amount of spacing between lines and Easy To Read being the most amount of spacing between lines.

Graphical user interface 43 is also provided with Preview button 60. By clicking on Preview button 60, a user can preview a first page of a formatted document in viewing area 61. An example of a previewed formatted document is shown in Figure 9A.

As shown in Figures 9A and 9B, WebFormatter also includes Options button 61. Options button 61 provides a user with additional formatting options which are used by WebFormatter to create a formatted do

整理番号=36P111J

(38)

document. A user can activate Options button 61 by clicking thereon. This causes Options dialog box 62, shown in Figure 9B, to appear on display 2.

As shown in Figure 9B, options dialog box 62 includes General options 64, Container options 66 and Strip Meta Info options 67. General options 64 includes "Text only" listbox 72, "Index of links in the page" listbox 73, and "No floating pictures" listbox 74. These options are indicated as being selected by a check mark or the like in a respective listbox. As will become clear from their descriptions, more than one of the options in General options 64 can be selected at the same time.

"Text only" listbox 72 instructs WebFormatter to strip all graphics in a Web page and print only text therein. "Index of links in the page" listbox 73 instructs WebFormatter to add a list of all URLs present in a Web page or pages to the end of a formatted document. Preferably, the list of URLs is printed as superscript, and anchor positions of the URLs in the list are marked in bold. "No floating pictures" listbox 74 instructs WebFormatter to print all images in the document in a particular area of the formatted document. In some cases, therefore, when this option is selected, WebFormatter shrinks images, as needed, so that images fit into a particular area.

Strip Meta Info options 67 provides engineering options which facilitate stripping of unnecessary information from a Web page being processed by WebFormatter. The options include (1) "None", which instructs WebFormatter to strip nothing from the Web page, (2) "Till the first horizontal rule", which instructs WebFormatter to strip all links and images until and up to predefined first and second horizontal formatting rules (e.g., up until a horizontal line across a page), and (3) "Till the first text", which instructs WebFormatter to strip all links and ima

整理番号 = 3 6 P 1 1 1 J(3 9)

ges up to first and last occurrences of text in the Web page. Only one of Strip Meta Info options 67 can be selected at a time. Selection thereof is indicated by a dot in a bullet located next to an option, as shown in Figure 9B.

Container options 66 provides options for processing documents, addresses for which are stored in container 76 shown in Figure 9B.

Prior to describing Container options 66, a description of container 76 will be provided.

As noted, container 76 stores URL addresses of selected documents. Document addresses which are input to field 44 are added to container 76. The order in which URLs are input into container 76 denotes the order in which data in the URLs is processed by WebFormatter. As shown in Figure 9B, once container 76 becomes full, its icon changes to that shown by reference numeral 77.

When a user clicks on the icon for container 76, menu 77 is displayed. Menu 77 provides five options; i.e., Open 79, Empty 80, Print 81, Edit 82 and Save 84. These options are highlighted when activated, and are described in detail below.

Open 79, when activated, displays Container Contents screen 87 shown in Figure 9B. Container Contents screen 87 shows the URL addresses stored in container 76. Container contents screen 87 provides four buttons; i.e., Add current URL button 88 which adds the current URL to container 76, Delete button 89 which permits a user to highlight and delete a URL in container 76, Empty button 90 which permits a user to empty container 76, and Done button 91 which permits a user to close Container Contents screen 87. It is noted that a user can also empty the contents of container 76 by clicking on Empty 80 of menu 77.

In addition, the user can rearrange the order of URLs stored in container 76 by dragging and dropping different URLs at different

整理番号 = 3 6 P 1 1 1 J

(4 0)

locations therein. As noted above, since the URLs are processed in the order that they appear in container 76, this feature permits a user to rearrange the processing order of the URLs in container 76 interactively.

Print 81, Edit 82 and Save 84, when activated, cause WebFormatter to download all data at Web pages defined by the URLs stored in container 76, format them as specified by the user, create RTF file(s) storing the formatted Web pages, and do the selected action, i.e., save, edit or print the RTF file(s). This process is described in greater detail below.

Referring back to Options dialog box 62, Container options 66 include "Print table of contents" listbox 92 and "Empty after processing" listbox 94. As shown, a check mark appears in a listbox to indicate that the listbox has been selected. In this regard, more than one listbox can be selected at a time. "Print table of contents" listbox 92, when selected, instructs WebFormatter to print titles of all URLs in container 76 as a table of contents in a formatted output document. "Empty after processing" listbox 94, when activated, instructs WebFormatter automatically to empty container 76 after printing, editing or saving a document, without waiting for a user to do so.

Also shown as part of Container options 62 are Select RTF Editor button 69, Cancel button 70 and OK button 71. By clicking on Select RTF Editor button 69, a user can select an RTF file editor, examples of which are noted above. This can be done, for example, by displaying another dialog box listing predefined RTF editors (not shown) and selecting one of the predefined RTF editors. Cancel button 70 cancels Container options 62 and OK button 71 confirms selected options in Container options 62 and then closes its dialog box.

As shown in Figure 9B, graphical user interface 43 also in

整理番号=36P111J

(41)

cludes print icon 96, edit icon 97, save icon 99, help button 100, done button 101 and minimizing icon 102. A user may select any of these features by clicking thereon using a mouse.

Print icon 96 opens a print dialog box (not shown), which allows a user to print any number of copies of Web pages formatted by WebFormatter. Edit icon 97 opens an RTF file storing formatted Web page(s) for editing by a predetermined RTF editor. Save icon 99 opens a save dialog box (not shown), which allows the user to name and save a formatted Web page as an RTF file. Help button 100 provides help messages for operating WebFormatter, and Done button 101 exits from WebFormatter. Minimizing icon 102 activates the minimizing mode of Webformatter which was mentioned above and which is described in greater detail below.

Figure 9C shows menus provided by WebFormatter during its operation. These menus include file menu 103, edit menu 104 and window menu 106. File menu 103 provides "Save", "Edit" and "Print" options, the functions of which are identical to those of Save icon 99, Edit icon 97 and Print icon 96, respectively. An "Exit" option is also provided to exit from File menu 103. Finally, File menu 103 provides "Open HTML file" option 107. This option provides a user with the capability to open a local HTML file; i.e., a hypermedia file resident on the user's computer such as a file saved from NetScape, or URL files created by dragging and dropping a URL onto the windows desktop. "Open HTML file" option 107 also provides hooks needed to open files created by other Web-file-processing products so that those files can be formatted as RTF files and printed, saved and/or edited using WebFormatter.

Edit menu 104 provides "Paste URL" option 109. "Paste URL" option 109 pastes the contents of a paste buffer, such as a URL address copied from a Web page, into URL field 44, as described above.

Window menu 106 provides a "Help Topics" option which prov

整理番号 = 3 6 P 1 1 1 J

(4 2)

ides a user with information regarding the use, maintenance and background of WebFormatter, and an "About WebFormatter" option which provides a user with a dialog box (not shown) containing WebFormatter's version number and copyright notice(s). Window menu 106 also includes "Preferences" option 110. "Preferences" option 110 opens preferences dialog box 112, shown in Figure 9D.

Preferences dialog box 112 is used to configure and re-configure WebFormatter. As shown in Figure 9D, preferences dialog box 112 includes Minimize view options 113, General options 114 and WWW Browser to use options 115. Minimize view options 113 can be set to configure WebFormatter's graphical user interface in the minimized mode. Two sets of options are provided. The first set include "Print", "Edit" and "Save". These options correspond to print icon 96, edit icon 97 and save icon 99, shown in Figure 9B. When a check mark appears in a listbox next to one of these options, the icon for that option is displayed in the minimized mode, e.g., the print icon, the edit icon and/or the save icon.

More than one option can be selected at once. In this regard, Figure 9H shows graphical user interface 116, which is a representative example of a graphical user interface for WebFormatter when WebFormatter is in the minimized mode.

Referring back to Figure 9D, Minimize view options 113 also include "Row" and "Stack" options. These options can be set to display WebFormatter's graphical user interface in the minimized mode horizontally by selecting "Row" or vertically by selecting "Stack".

Only one of these options can be selected at a time. As an example of the foregoing, graphical user interface 116 corresponds to a row of icons.

WWW Browser to use options 115 determine which World Wide Web browser is to be used with WebFormatter. As shown, preferably NetSc

整理番号 = 3 6 P 1 1 1 J(1 3)

ape, Internet Explorer and Mosaic are provided as browser options; however, other browser options can also be provided. As might be expected, only one of these options can be selected at a time. The default browser option is NetScape Navigator.

General options 114 include "Auto-start with browser" option 117, "Open in minimized view" option 118, "Warn before printing more than __ pages" option 119, and "Warn before saving more than __ MBs" option 120. "Auto-start with browser" option 117 sets WebFormatter to be invoked automatically when a Web browser is activated. If this option is not selected (which is the default), WebFormatter is opened by double clicking on a WebFormatter icon in the windowing environment, selecting WebFormatter from the Windows start menu, or dragging and dropping a URL from the Web browser into the WebFormatter icon, as described in more detail above. "Open in minimized view" option 118, when selected, opens WebFormatter in minimized mode. The default, however, is the fully-functional mode. "Warn before printing more than __ pages" option 119, and "Warn before saving more than __ MBs" option 120 allow a user to control the number of pages saved of a formatted document and the amount of memory space used by those pages, respectively. The default for both of these options is for no warning to be given. As is evident, more than one of the general options can be selected at the same time.

Preferences dialog box 112 also includes cancel button 121 which cancels a user's selected preferences and OK button 122 which confirms a user's selected preferences.

As explained above, WebFormatter can be configured to enter directly into the minimized mode via Preferences dialog box 112, or a user can enter the minimized mode via minimizing icon 102 shown in Figure 9B. As also noted above, Figure 9B shows an example of graphical user interface 116 for WebFormatter in the minimized mode. Graphical user i

整理番号-36P111J(44)

nterface 116 is displayed as a floating interface while a user is exploring the Web. Thus, as a user views a Web page, the user also views graphical user interface 116. By clicking on an appropriate icon on graphical user interface 116 (which, in Figure 9E, includes icons identical in both structure and function to those shown in graphical user interface 43), the user can capture the current Web page, process and format the Web page into an RTF file, and save, edit and/or print the RTF file. Alternatively, the user can drag a URL from the Web browser and drop it into one of the icons.

A user can reconfigure WebFormatter in the minimizing mode by double clicking a right mouse button. This action causes a preferences dialog box to appear on display 2 which is identical to preferences dialog box 112. Thereafter, the user can alter the configuration of WebFormatter as desired. Should a user wish to enter the fully-functional mode from the minimizing mode, the user need merely click on maximizing icon 117 shown in Figure 9E.

Figure 10 is a flow diagram describing the operation of WebFormatter. WebFormatter is activated in step S1000. As described above, this can be done by double-clicking on a WebFormatter icon in a windowing environment. Depending upon how WebFormatter has been configured, i.e., in the fully-functional mode or the minimizing mode, either a graphical user interface similar to that of graphical user interface 43 or one similar to that of graphical user interface 116 is displayed in step S1000. For the sake of completeness, the following assumes that a graphical user interface similar to that of graphical user interface 43 is displayed in step S1000, since the default mode of WebFormatter is the fully-functional mode.

Next, in step S1001, WebFormatter is configured, as described above via preferences dialog box 112 and options dialog box 62. Thi

整理番号 = 3 6 P 1 1 1 J

(4 5)

s step is not necessary unless a user wishes to change WebFormatter's previously set configuration. In step S1002, document format data is input in fields 44 and 46 to 49 described above. More specifically, the user inputs a URL, (or filename in alternative embodiments) into URL field 44. As described below, WebFormatter uses this information to process Web pages stored at the URL to create an RTF file based on the configuration of WebFormatter and the data input in fields 46 to 49.

In step S1003, a Web reader similar to that of Web reader 34 described above is executed. The Web reader connects to a network, such as the World Wide Web, in step S1004.

Next in step S1005, it is determined whether a URL or a filename has been entered. As described above, in preferred embodiments of WebFormatter, only a URL may be entered. However, since alternative embodiments of WebFormatter may permit entry of a filename, a description of processing a file other than one at a URL address will be provided.

If a URL has been entered in field 44, processing proceeds to step S1006. In step S1006, the Web reader accesses the hypermedia document (e.g., a homepage) specified by the URL address. In step S1007, WebFormatter instructs the Web reader to traverse the hypermedia document. Thereafter, WebFormatter selects URL address(es) from the Web and stores the addresses in container 76. Once all desired addresses have been selected and a processing function, such as print, has been activated, WebFormatter downloads data stored at the addresses in container 76 in to memory 5. WebFormatter then extracts predetermined data from the downloaded data based on the configuration information set in Optional dialog box 62, and stores the extracted data in memory 5. Thus, for example, if "Text Only" option 72 in Options Window 62 is on, only text is extracted from the downloaded data. Processing then proceeds to step S1011.

On the other hand, if, in step S1005, a filename for an HT

整理番号=36P111J(46)

ML source file is entered, WebFormatter instructs the Web reader to access a first site in the file. In steps S1008 and S1009, the site is traversed and data is extracted and stored in the same manner as in step S1007, described above. Then, in step S1010, WebFormatter determines if more sites are listed in the HTML source file. If more sites are listed in the file, flow returns to step S1008, and the next site is accessed. If no more sites are present, processing proceeds to step S1011.

In step S1011, WebFormatter processes the extracted data in accordance with the previously set format information. For example, if Columns field 48 is set to multiple, the extracted data will be formatted into a document having multiple columns. The above processing is initiated by activating one of Print icon 96, Edit icon 97 or Save icon 99, and is similar to the processing described above in the first embodiment, e.g., flattening the document and formatting the document based on the formatting information. Accordingly, a detailed description thereof is omitted for the sake of brevity.

Once the documents whose URLs are stored in the container have been downloaded, formatted according to the preset formats and configurations, and converted into RTP file(s) in step S1011, in step S1012, the RTP file(s) are output. Alternatively, the RTP files(s) can be edited or saved, depending upon which icon on the graphical user interface has been activated.

The invention has been described with respect to particular illustrative embodiments. It is to be understood that the invention is not limited to the above described embodiments and modifications thereto, and that various changes and modifications may be made by those of ordinary skill in the art without departing from the spirit and scope of the appended claims.

APPENDIX 1

整理番号=36P111J(47)

SAMPLE USER PROFILE

The User Profile is implemented in windows.ini file format.

[Defaults]

Count=4

Title=My Daily Paper

[1]

Heading=News In Brief

Site=1

Section=Front Page

MaxLevels=5

MaxPages=10

MaxKBytes=2000

Date=today

Print=level 0

Template=1

[2]

Heading=Sports In Brief

Site=2

Section=Sports

Max Levels=0

MaxPages=10

MaxKbytes=200

KeywordFilter="Football" AND "49ers"

Date=today

Print=level 0

Template=1

[3]

Heading=Money Matters

Site=1

整理番号-36P111J(48)

Section=Business

MaxLevels=1

MaxPages=100

MaxKBycs=20000

KeywordFilter="Computer" OR "hardware" OR "Software"

Date=today

Print=all

Template=2

[4]

Heading=Sri Lanka

Site=3

Section=HotNews

MaxLevels=1

MaxPages=100

MaxKBytes=20000

Date=today

Print=leaves

Template=2

SAMPLE SITE PROFILES

#Legend:

#%W-day of the week

#%s-section part of URL

[Defaults]

Count=3

[1]

Title=San Jose Mercury News

Username=mwickram

Password=cannon

StartData=StartHeadlines

整理番号 = 3 6 P 1 1 1 J

(4 9)

EndData=EndHeadlines

Home Page=http://www.sjmercury.com/

SectionURL=http://www.sjmercury.com/%S.htm

SectionCount=9

Section 1=Front Page

Section 2=International

Section 3=National

Section 4=Local & State

Section 5=Editorials Commentary

Section 6=Business

Section 7=Sports

Section 8=Living

Section 9=Entertainment

[1.Sections]

Front Page=front

International=intl

National=natl

Local & State=loc

Editorials & Commentary=edit

Business=biz

Sports=spts

Living-liv

entertainment=ent

[2]

Title=The San Francisco Chronicle

Home Page=http://www.sfgate.com/chronicle/

SectionURL="http://www.sfgate.com/cgi-bin/chronicle/article
list.cgi?%S:/chronicle/today"

整理番号=36P111J

(50)

Section Count=5

Section 1=News

Section 2=Business

Section 3=Sports

Section 4=Editorial

Section 5=Datebook

[2, Sections]

News=News:MN

Business=Business:BU

Sports=sports:SP

Editorial=Editorial:ED

Datebook=Datebook:DD

[3]

Title=The Day News

Home page=

<http://www.lanka.net/lakehouse/ancWeb/dailynew/>

SectionURL="http://www.lanka.net/lakehouse/ancWeb/dailyne

w/%W/WS.html"

SectionCount =12

Section 1=Business

Section 2=Editorial

Section 3=Features

Section 4=Foreign

Section 5=Letters

Section 6=InBrief

Section 7=HotNews

Section 8=Probes

Section 9=Military

Section 10=Politics

整理番号=36P111J

(51)

Section 11=Obituaries
 Section 12=Sports
 [3. Sections]
 Business=business/intro
 Editorial=editorial/final
 Features=features/intro
 Foreign=foreign/intro
 Letters=letters/final
 InBrief=inbrief/intro
 HotNews=hotnews/intro
 Probes=probes/intro
 Military=military/intro
 Politics=politics/intro
 Obituaries=obiturai/intro
 Sports=sports/intro

APPENDIX 2

SYNTAX FOR RETRIEVAL, EXTRACTION AND PRINTING CRITERIA

Maximum levels to search: MaxLevels=<#>

-1: to retrieve all levels

0-n: to retrieve up to n levels

Maximum pages of the document: MaxPages=<#>

n: final document not more than n pages

Maximum size of the document: MaxKBytes=<#>

n: document size not more than n kilo bytes

Exclusion rules:

Date=today|lessthan <#>

today: retrieve only articles posted today

lessthan <#>;n: retrieve only articles no
 more than n days old

整理番号=36P1111J

(52)

Retrieve=all|nosubdir|nothisdir|thissiteonly

all: allow to fetch pages from other sites

nosubdir: exclude URLs to subdirectories

nothisdir: exclude URLs in this directory

thissiteonly: fetch pages from this site only

y

Keyword search:

KeywordFilter-<keyword> (AND|OR|NOT) <keyword>:

accumulate only pages containing the combination of keywords

KeywordRank-<#>;n: use fuzzy logic to rank

pages according to keyword combination in KeywordFilter and keep top n ranked pages

KeywordAuthor=<author>: accumulate only

pages authored by author

ExcludeType=ads|nonEnglish

ads:exclude advertisements

nonEnglish:exclude articles that are not in

English

Flattening rules: Print=all|leaves|level-<#>

all: include all nodes in the tree in the linear document

leaves: include all leaves in the tree in the linear document

level=<#>;n: include up to nth level of the tree in the linear document

Formatting rules: Template-<#>

n: print according to default or user template number n

整理番号 = 3 6 P 1 1 1 J

(5 3)

APPENDIX 3

DESCRIPTION OF MODULES

Appendix 3A

THE PERSONAL NEWS PROFILE EDITOR MODULE

The Profile Editor manages access to the user profiles and is represented by CProfileMgr class. It also manages loading and saving of the profiles. The services provided by Profile Editor are:

BOOL NewProfile(CString fileName);

- Creates a new profile given the file name.

BOOL OpenProfile();

- Opens the default profile.

BOOL OpenProfile(CString fileName);

- Opens the named profile.

CProfileEntry* GetFirstEntry();

- Loads and returns the next profile entry.

CProfileEntry* GetNextEntry();

- Loads and returns the next profile entry.

BOOL WriteEntry(CProfileEntry& entry);

- Saves a new entry in the profile.

Each profile entry contains an extraction specification and an output specification as represented by CProfileEntry class. The methods provided are:

CURL GetSiteId();

- Returns the site id contained in the profile entry.

整理番号 = 36P111J(54)

CExtractionSpec GetExtractionSpec();

- Returns the extraction specific
ation contained in the profile entry. Extraction specification contains
keywords for searching, limits for levels, pages, size in kilo bytes.

COutputSpec GetOutputSpec();

- Returns the output specificatio
n contained in the profile entry. Output specification contains formatt
ing instructions and tree traversal rules.

THE Web READER MODULE

CWebPage class abstracts the interface to the Internet browser and is re
presentative of the actual Web page. It will be responsible for fetchin
g a Web page, extracting links or references to other URLs in the Web pa
ge, and maintaining the contents of a Web page. The methods provided ar
e:

BOOL Load();

- Fetch the Web page using the UR
L, username and password.

BOOL Parse();

- Parses the data in the Web page
and creates a list of links. Also resolves the relative URLs into abso
lute URLs.

CURLList* GetLinks();

- Returns the list of links in th
e Web page.

CPageData* GetData();

- Returns the actual text data co
ntained in the Web page.

void FilterContent();

- Extracts title and other inform

整理番号=36P111J

(55)

ation according to the site data.

CString GetTitle();

-

Returns title and other informa

tion according to the site data.

CString GetAuthor();

-

Returns the author of the Web p

age.

int GetSize();

-

Returns the size of the data in

kilo bytes.

CNetwork class will encapsulate OLE functionality and provides communica
tion with the Internet browser.

CString GetUsername();

-

Determine the currently set use

rname.

void SetUsername(LPCTSTR);

-

Set the current username in the

CNetwork object.

CString GetPassword();

-

Determine the currently set pas

sword.

void SetPassword(LPCTSTR);

-

Set the current password in the

CNetwork object.

void Close();

-

Disconnect any active connectio

n and reset the CNetwork object.

short Read(BSTR* pBuffer, short Amount);

-

Read data retrieved by the Brow

整理番号 = 3 6 P 1 1 1 J(5 6)

ser.

long GetStatus();

- Query the status of the current

load.

BOOL Open(LPCTSTR pURL, short Method, LPCTSTR pPostData, long lPostDataSize, LPCTSTR pPostHeaders);

- Initiates the retrieval of a UR

L from the network.

CString GetErrorMessage();

- Provide the caller with interna

lly generated error messages.

short GetServerStatus();

- Determine the error status repo

rted by the server.

long GetContentType();

- Return the content length (total

amount of bytes) of the current load.

CString GetContentEncoding();

- Return the MIME encoding of the

current load.

CString GetExpires();

- Return when the data retrieved

by this load is no longer considered valid.

CString Resolve(LPCTSTR pBase, LPCTSTR pRelative);

- Generate an absolute (fully qua

lified) URL.

BOOL IsFinished();

- Determine if a load is complete

整理番号=36P111J

(57)

```
short BytesReady();
```

Inform the caller of the number of bytes prepared to be read.

THE SITE DRIVER MODULE

The Site Driver will provide the site information to the Web Reader. The Site Driver is functionally similar to the Profile Editor and is represented by CSiteDriver class. Services provided are:

```
BOOL NewProfile(CString fileName);
```

- Creates a new profile given the file name.

```
BOOL OpenProfile();
```

- Opens the default profile.

```
BOOL OPenProfile(CString fileName);
```

- Opens the named profile.

```
CSiteProfile* GetFirstSite();
```

- Loads and returns the first site entry.

```
CSiteProfile* GetNextSite();
```

- Loads and returns the next site entry.

```
BOOL WriteEntry(CSiteProfile& entry);
```

- Saves a new entry in the profile.

```
int NumberOfSites();
```

- Returns the number of sites specified in the profile.

An entry in the site profile will contain information about the base URL of the site, title of the news source, information about how to access the site, and various other information such as section data etc. and wi

整理番号=36P111J

(58)

It is represented by CSiteEntry class. Methods provided are:

CString GetURL();

- Returns the base URL of the site.

e.

CString GetUsername();

- Returns the username for the site.

te.

CString GetPassword();

- Returns the password for the site.

te.

CString GetTitle();

Returns the password for the site.

te.

CString GetTitle();

Returns the title of the news source.

source.

int SectionCount();

- Returns

Appendix 3B

TREE MANAGER MODULE

Tree Manager will maintain the most central data structure in this program, which is a tree of Web page nodes and is represented by the CPageTree. CPageTree will traverse the WWW to retrieve the necessary Web pages according to the extraction specification and builds the tree. The methods provided are:

CPageTreeNode* GetRoot();

- Returns the root node of the tree.

ee.

BOOL Build(CURL URL, CExtractionSpec& spec);

整理番号 = 3 6 P 1 1 1 J(5 9)

- Builds the tree according to the personal news profile extraction specification.

Each node in the page tree is represented by a CPageTreeNode. Methods provided are:

BOOL AddChild(CWebPage* page);

- Adds a child node with Web page data.

CWebPage* GetPage();

- Returns the Web page contained in the node.

int NumberOfChildren();

- Returns the number of children belonging to the node.

BOOL IsLeaf();

- Returns TRUE if a leaf node, i.e., no children.

To traverse the Web page tree, a CTrecIterator class is defined with different traversal methods. Methods provided are:

void Reset();

- Cancels the current transferral, and initializes state data.

CPageTreeNode* GetNextNode();

- Returns the next node in the tree in a depth first search.

CPageTreeNode* GetNextSibling();

- Returns the next node in the tree in a breadth first search.

CPageTreeNode* GetNextLeaf();

- Returns the next leaf in the tree.

整理番号=36P111J(60)

ee in a depth first search.

THE FORMATTER MODULE

Input to this module will be the Web page tree created by the Tree Manager and the output specification contained in the user profile. Formatter will traverse the tree according to the rules specified in the output specification and the final document will be formatted using the formatting instructions in the output specification and the formatting contained in the Web pages such as headings, paragraphs and lists etc.

The output document will be in Rich Text Format (RTF) and will be accessible by many applications. RTF is a advanced formatting language for text, providing document, section and paragraph formatting, style sheets, headers and footers, and with support for Unicode. Image formats supported are DIB, DDB, WMF, OS/2 metafiles. There is no support for Web images which are of the GIF format. A third party library will need to be purchased in order to do the conversion of the GIF to DIB format or one can be developed in-house.

The prototype creates a HTML file as the output.

The formatter is represented by the CFormatter class. The methods provided are:

```
BOOL OpenHTMLFile(CString fileName);
```

- Opens the named HTML file for output.

```
void CloseHTMLFile();
```

- Closes and saves the HTML file.

```
BOOL PrintHTML(CPageTree& root, COutputSpec& format);
```

- Given the root and the output specification, traverses the tree and prints the contents in the Web pages in HTML format.

```
BOOL OpenRTFFile(CString fileName);
```

整理番号=36P111J

(61)

- Opens the named RTF file for output.

void CloseRTFFile();

- Closes and saves the RTF file.

BOOL PrintRTF(CPageTree& root, COutputSpec& format);

- Given the root and the output specification, traverses the tree and prints the contents in the Web pages in RTF format.

BOOL Print(CPageTree& root, COutputSpec& format);

- Given the root and the output specification at, traverses the tree and prints the contents in the Web pages to the default printer.

4. BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a perspective view showing the outward appearance of the personal news retrieval system according to the invention.

Figure 2 is a block diagram of the personal news retrieval system shown in Figure 1.

Figure 3, comprised of Figures 3A, 3B, 3C and 3D, shows representational diagrams illustrating an example of the transformation of information from the Web (Figure 3A) to an extracted data tree (Figure 3B), then to a flattened document (Figure 3C), and finally to a formatted document (Figure 3D) according to the invention.

Figure 4 is a representational block diagram of the manner by which a personal-news-profile for retrieving news articles via the Web is created or edited according to the invention.

Figure 5, comprised of Figures 5A and 5B, shows flow diagrams describing how a personal-news-profile is created or edited.

Figure 6 is a representational block diagram of the manner by which news articles are retrieved from the Web and formatted with re

整理番号 = 3 6 P 1 1 1 J

(6 2)

ference to a personal-news-profile according to the invention.

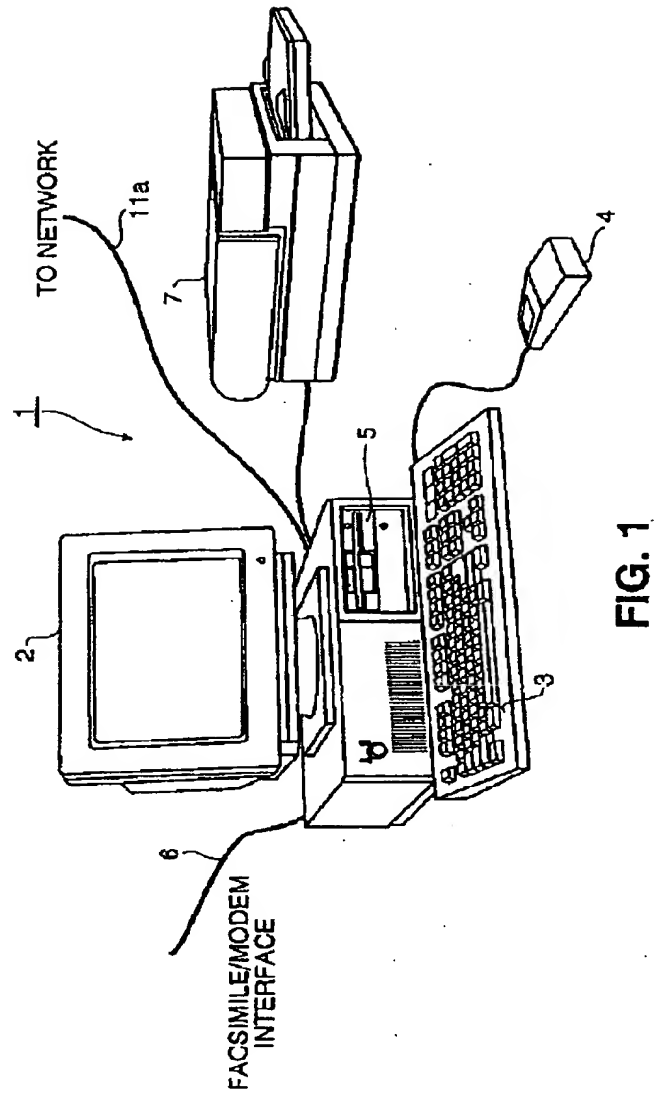
Figure 7 is a flow diagram describing how news articles are retrieved from the Web with reference to a personal-news-profile.

Figure 8 is a flow diagram showing how retrieved news articles are formatted with reference to a personal news profile and sent to a print device interface.

Figures 9A to 9B depict a graphical user interface used with the second embodiment of the present invention.

Figure 10 is a flow diagram describing the operation of the second embodiment of the invention.

【例 1】



整理番号=36P111J

(2)

【図2】

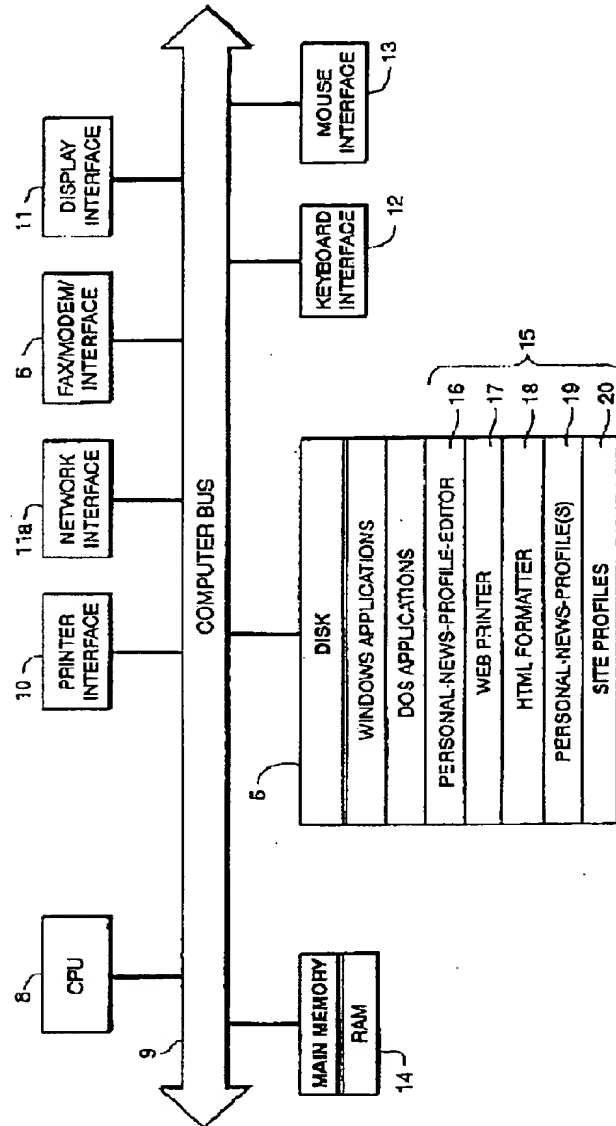


FIG. 2

整理番号=36P1111J

(3)

【図3A】

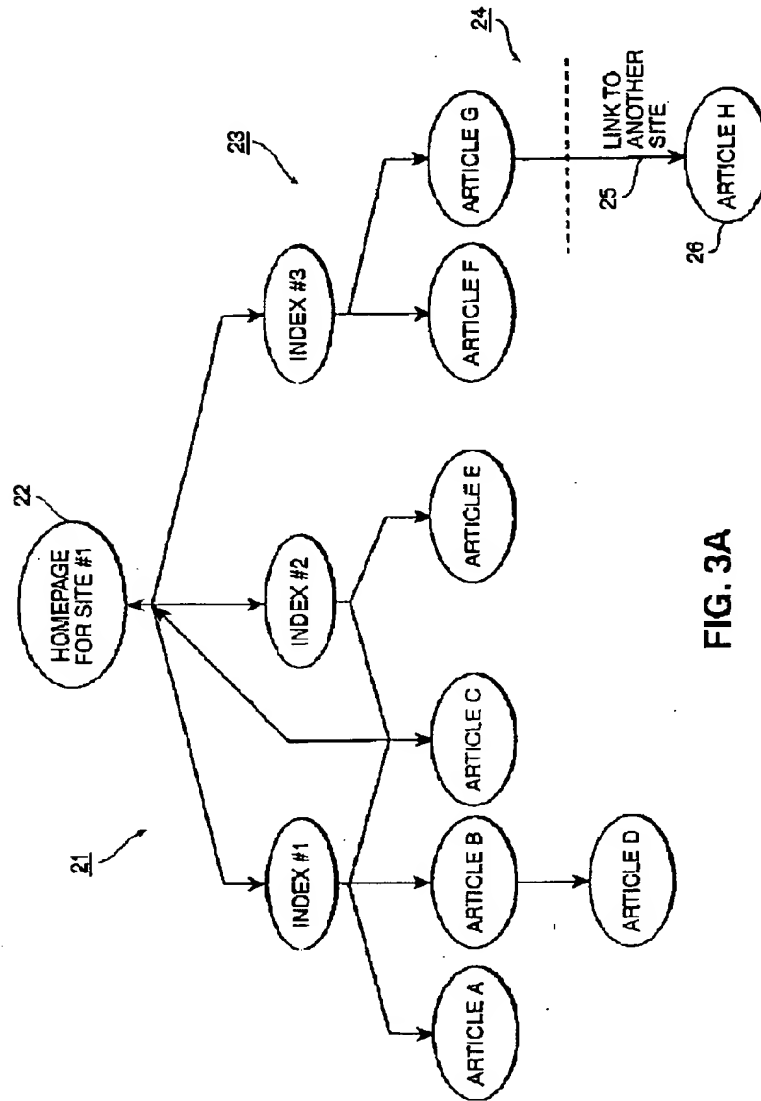


FIG. 3A

整理番号 = 36P111J

(4)

【図3B】

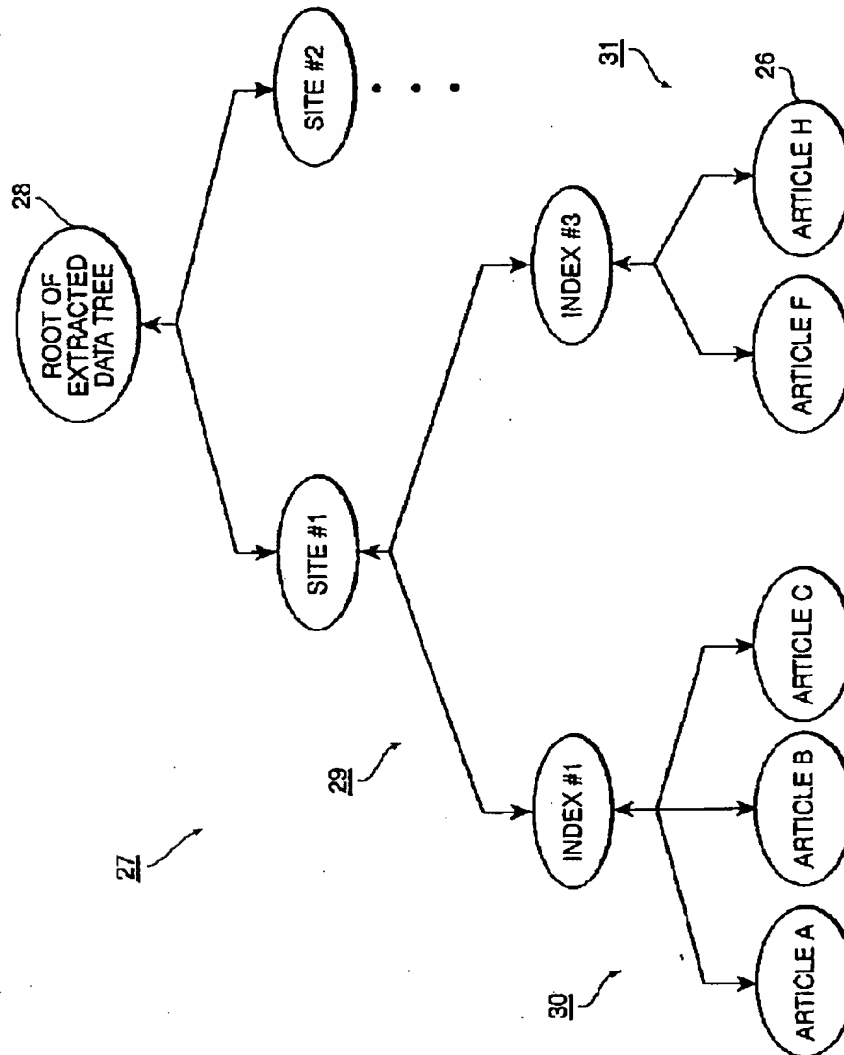


FIG. 3B

... (5) ...

FIG. 3C

FIG. 3D

FIG. 3D

MY PAPER	
From the <SITE #1 NAME>: Index #1 <ARTICLE A TITLE> This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the : : : <ARTICLE H TITLE> This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H. : : :	
From the <SITE #2 NAME>: Index <ARTICLE X TITLE> This is the text from article X. This is the text from article X. This is the text from article X. This is the text from : :	

整理番号 = 36P111J

(6)

【図4】

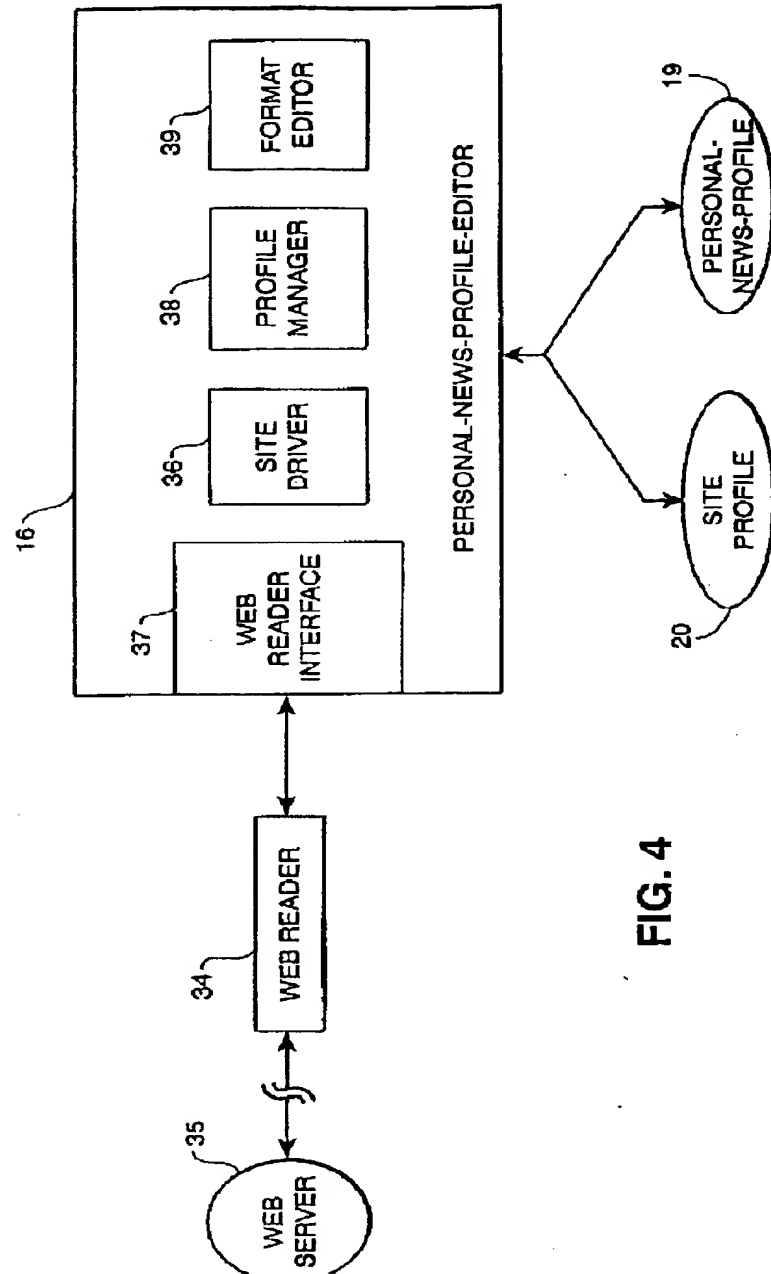


FIG. 4

整理番号=36P111J

(7)

【図5A】

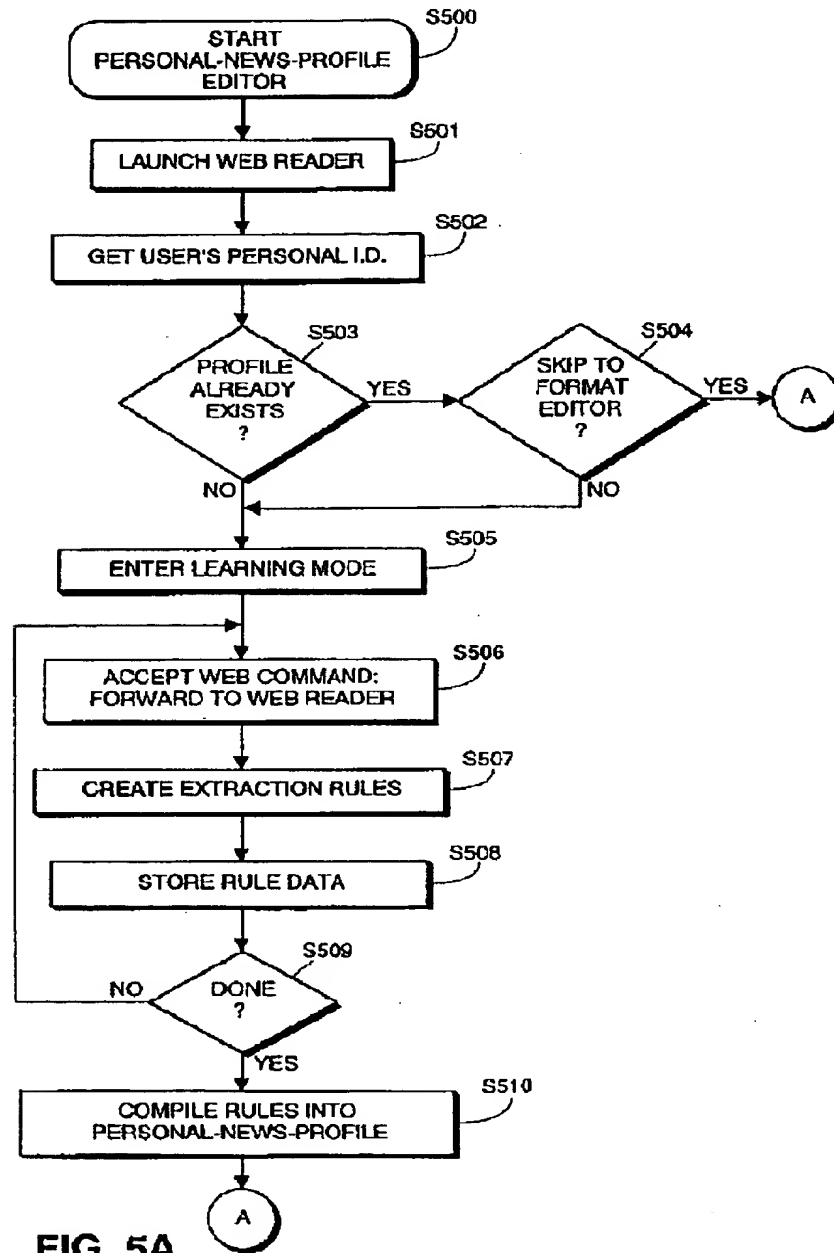


FIG. 5A

整理番号=36P111J

(8)

【図5B】

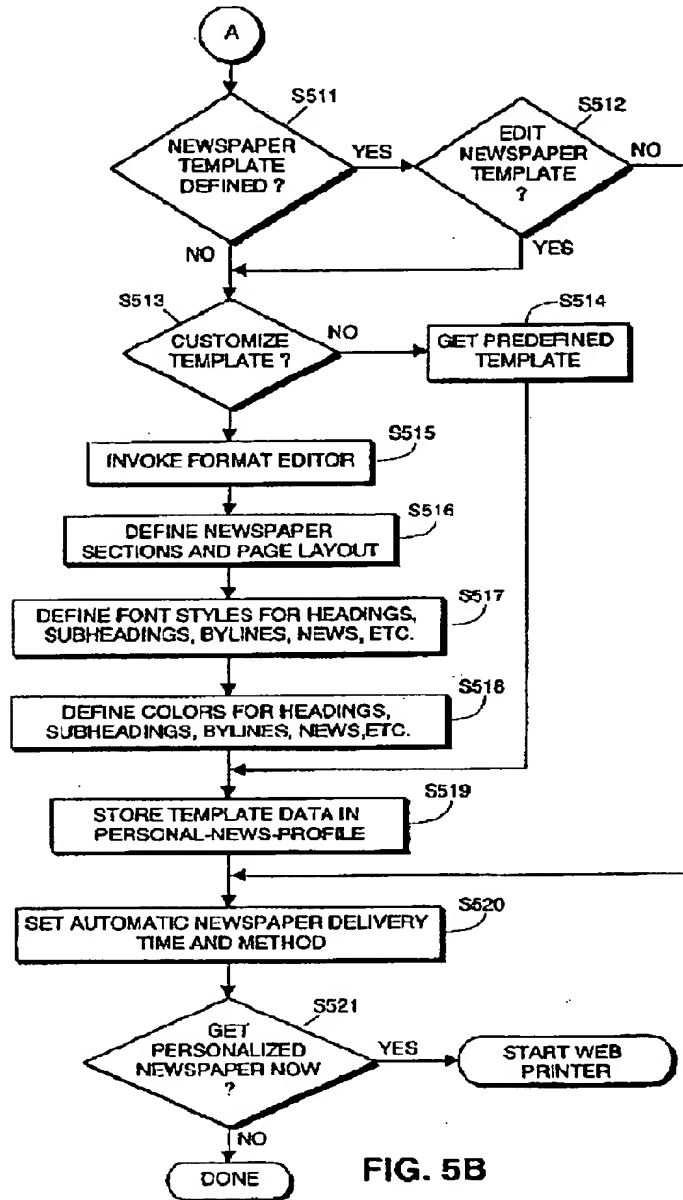


FIG. 5B

整理番号=36P111J

(9)

【図6】

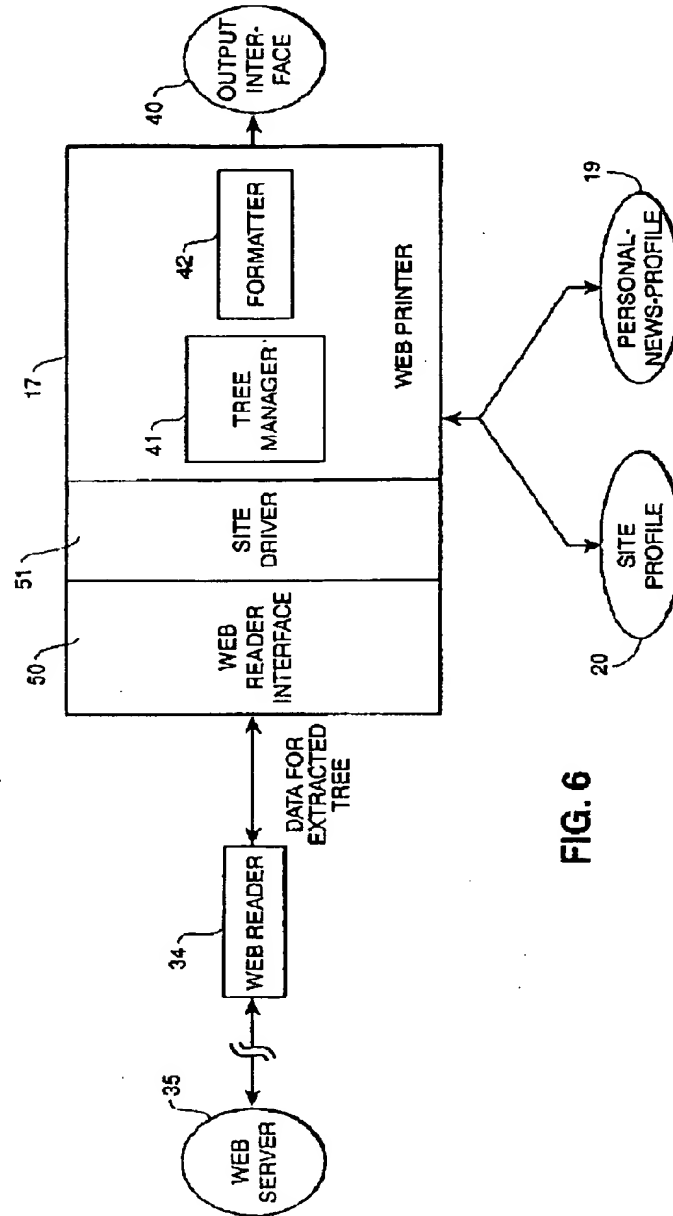


FIG. 6

整理番号=36P111J

(10)

【図7】

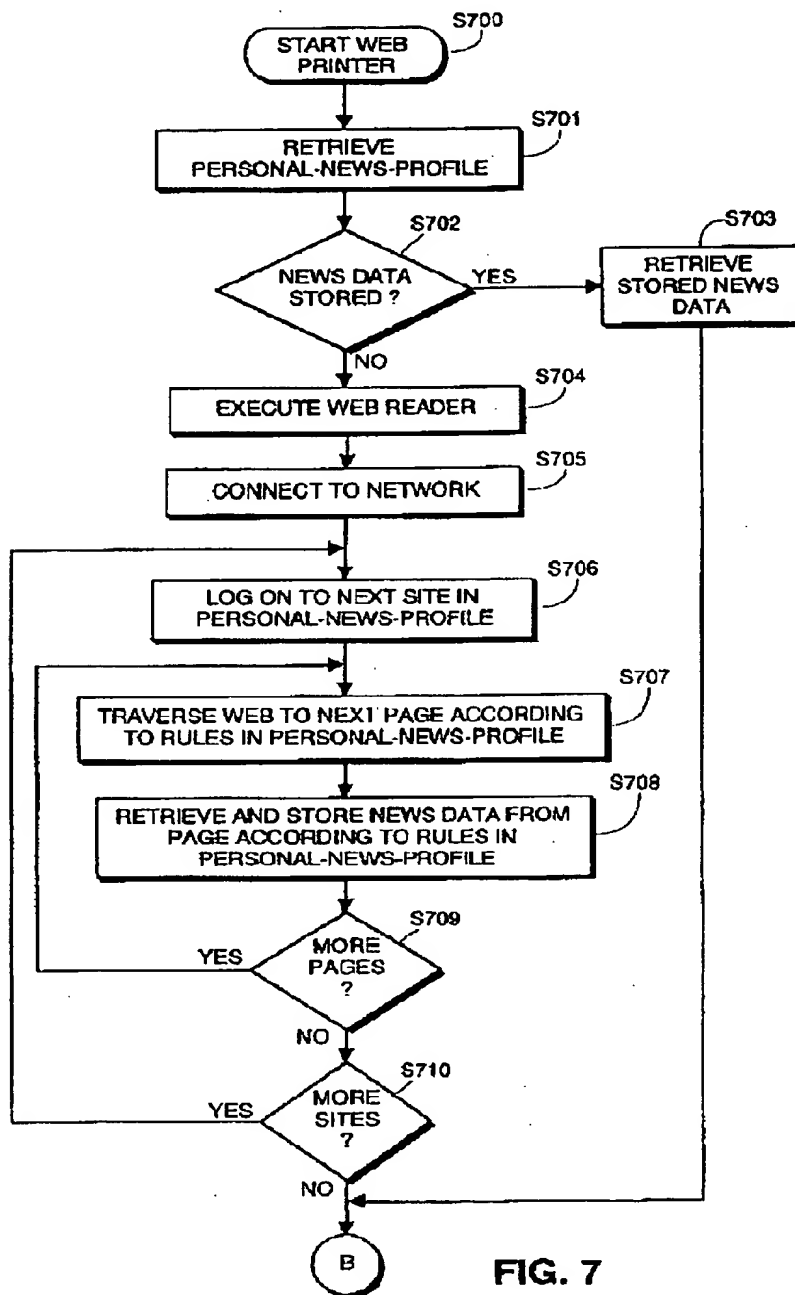


FIG. 7

整理番号=36P111J

(11)

【図8】

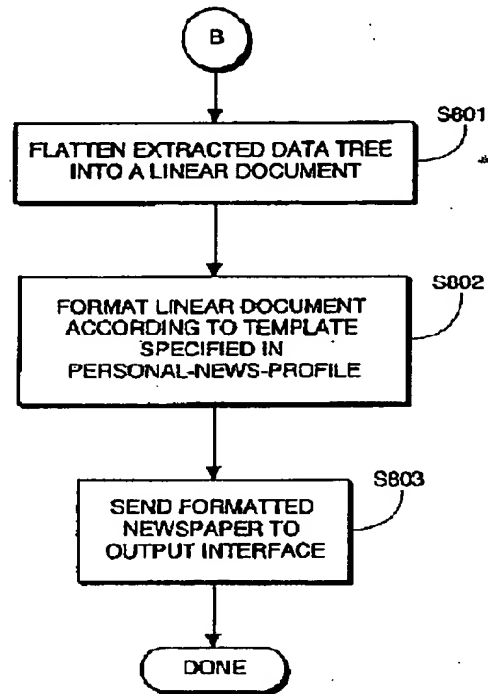


FIG. 8

整理番号=36P111J

(12)

【図9A】

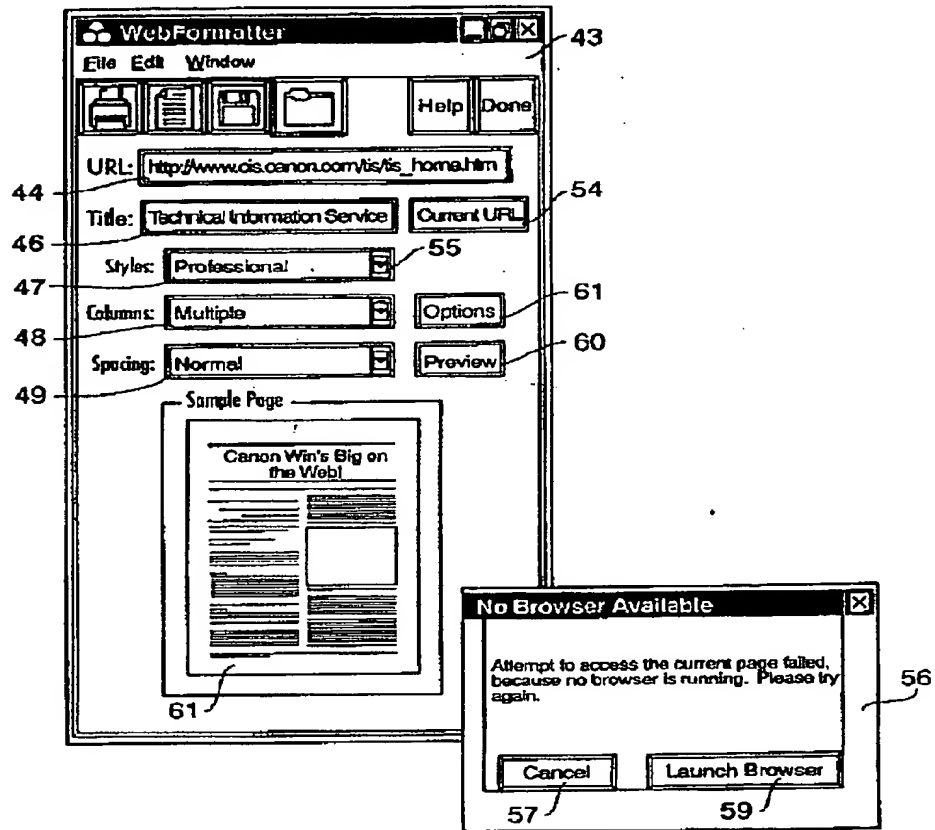


FIG.9A

整理番号 = 36P111J

(13)

【図9B】

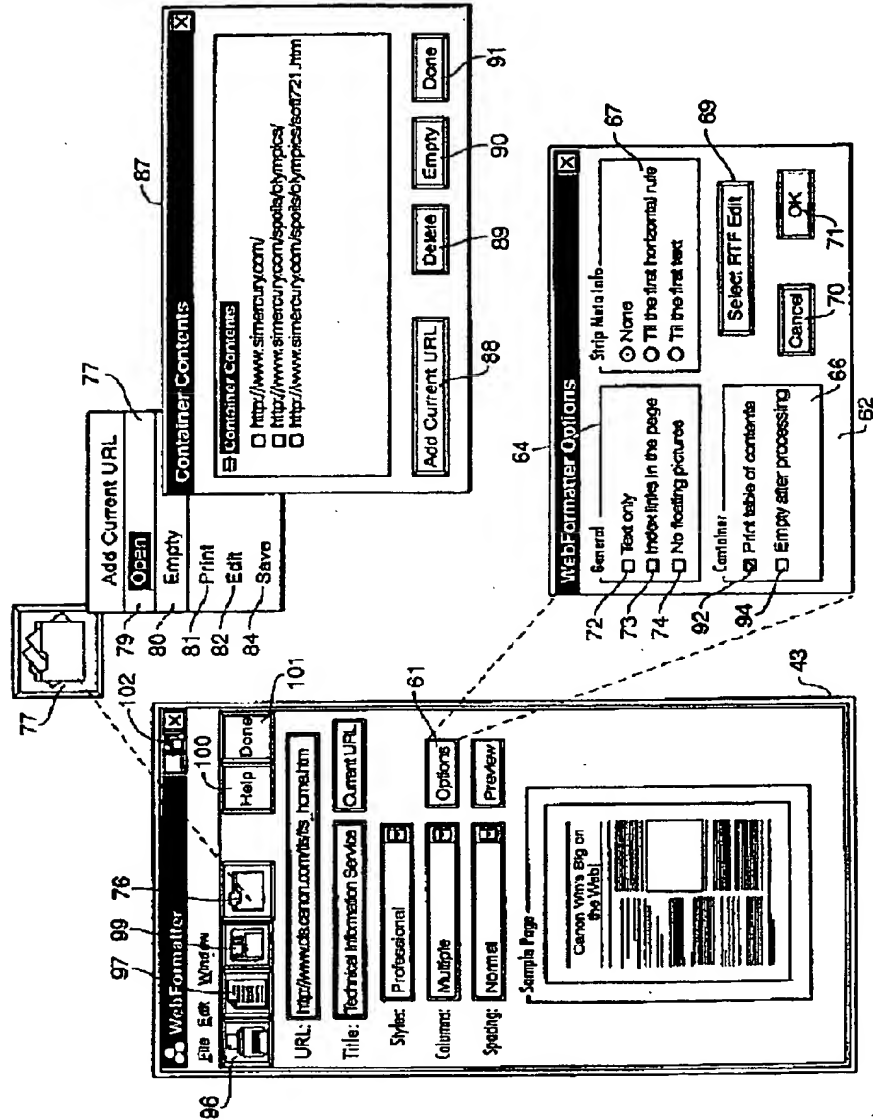


FIG.9B

整理番号=36P111J

(14)

【図9C】

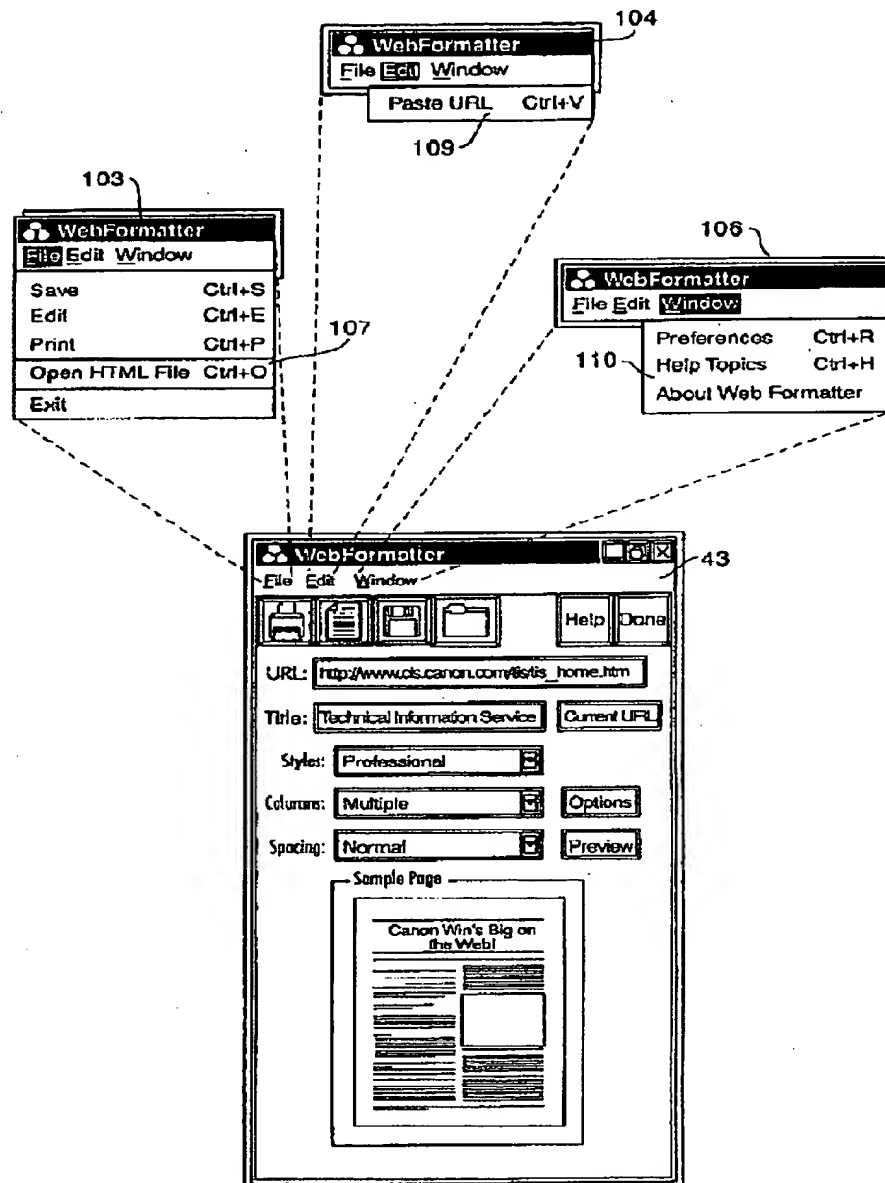


FIG.9C

整理番号=36P111J

(15)

【図9D】

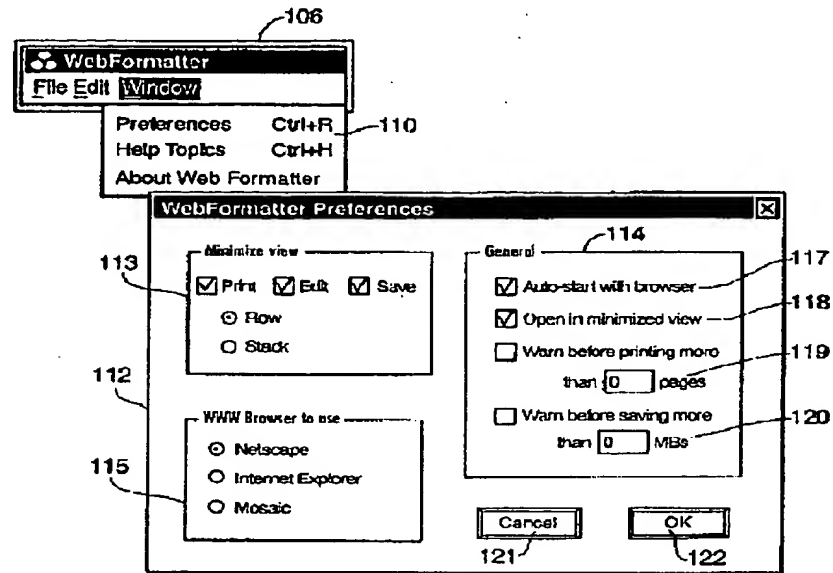


FIG.9D

【図9E】

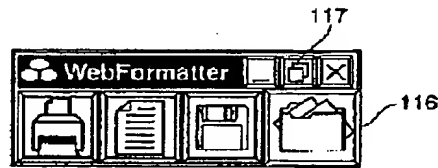


FIG.9E

【図10】

整理番号=36P111J

(16)

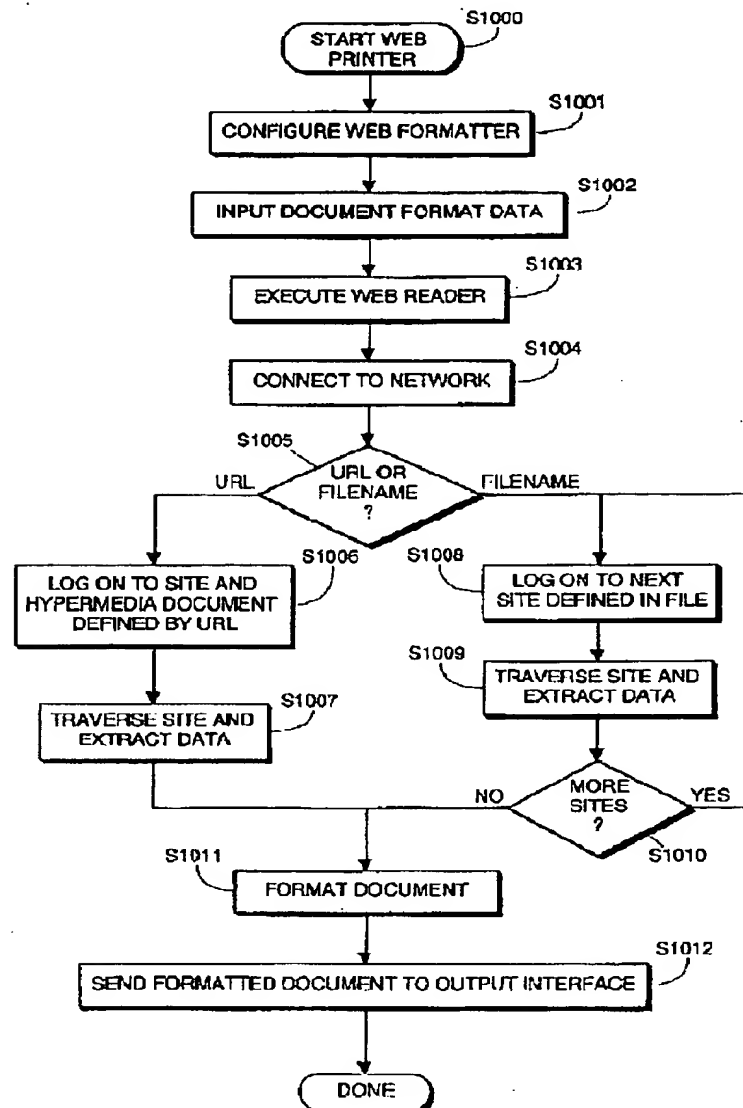


FIG. 10

1

ABSTRACT

A World Wide Web site data retrieval system includes an input device for inputting data and commands to access the World Wide Web, and a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information. The memory also stores process steps to connect to a Web site and to issue commands within the connected Web site, and a connection to the World Wide Web. The system includes a processor for launching the Web site data retrieval driver in response to a command to access the World Wide Web. The Web site retrieval driver, upon being launched, (1) launches the Web reader to connect to the World Wide Web via the connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into a personalized document based on the format information.

2

REPRESENTATIVE DRAWING

Fig. 3A